கணிப்பொறி இயல்

மேல்நிலை – இரண்டாம் ஆண்டு

கருவிகள் & பொருள்நோக்குத் தொழில்நுட்பம்

தமிழ்நாடு அரசு
இலவசப் பாடநூல் வழங்கு
திட்டத்தின் கீழ் வெளியிடப்பட்டது
(விற்பனைக்கு அன்று)

தீண்டாமை ஒரு பாவச்செயல்
தீண்டாமை ஒரு பெருங்குற்றம்
தீண்டாமை மனிதத் தன்மையற்ற செயல்

தமிழ்நாடு பாடநூல் மற்றும்
கல்வியியல் பணிகள் கழகம்
கல்லூரிச் சாலை, சென்னை – 600 006.

© தமிழ்நாடு அரசு
முதல்பதிப்பு – 2006
மறு பதிப்பு – 2017

தலைவர், பாடத்திட்டக் குழு
முனைவர் பாலகுருசாமி இ, முன்னாள் துணைவேந்தர்,
அண்ணா பல்கலைக் கழகம், சென்னை.

பாடப் புத்தக ஒருங்கிணைப்பாளர்
முனைவர் சங்கர நாராயணன் வி, இயக்குனர்,
தமிழ் இணையப் பல்கலைக் கழகம், சென்னை.

ஆங்கில மூல நூலாசிரியர்
முனைவர் கோபால் டி வி, அண்ணா பல்கலைக் கழகம், சென்னை.
முனைவர் மு. பொன்னவைக்கோ, SRM நிகர்நிலைப் பல்கலைக் கழகம்,
சென்னை.
திருமதி வசந்தி கிருஷ்ணகுமார், சைஃபி லிமிடெட், சென்னை.
திருமதி ருக்மணி கே, பத்ம சேஷாத்திரி பாலபவன், சென்னை.
முனைவர் சங்கர நாராயணன் வி, இயக்குனர்,
தமிழ் இணையப் பல்கலைக் கழகம், சென்னை.

மீள்பார்வையாளர்
முனைவர் கோபால் டி வி, அண்ணா பல்கலைக் கழகம், சென்னை.
திருமதி வசந்தி கிருஷ்ணகுமார், சைஃபி லிமிடெட், சென்னை.

அமைப்புப் பதிப்பாளர்
திருமதி சுபா ரவி, இயக்குநர், டிஜிடாடி கன்சல்டன்ஸி பிரைவேட்
லிமிடெட், சென்னை.

நூலாசிரியர்
முனைவர் மு. பொன்னவைக்கோ, SRM நிகர்நிலைப் பல்கலைக் கழகம், சென்னை.
சிவலிங்கம் மு, பொறியாளர், பி.எஸ்.என்.எல்., சென்னை.

விலை: ரூ.

பாடங்கள் தயாரிப்பு : தமிழ்நாடு அரசுக்காகப்
பள்ளிக் கல்வி இயக்ககம், தமிழ்நாடு

இந்நூல் 70 ஜி.எஸ்.எம். தாளில் அச்சிடப்பட்டுள்ளது.

ஆப்செட் முறையில் அச்சிட்டோர் :

முன்னுரை

நாம் தகவல் புரட்சி யுகத்தில் வாழ்ந்து கொண்டிருக்கிறோம். இந்தப் புரட்சியின் உந்து சக்திகள், கணிப்பொறியும் தகவல் தொடர்புத் தொழில்நுட்பங்களும் ஆகும். தனி நபர்களும் நிறுவனங்களும் இன்றைக்குத் தங்களின் அன்றாடப் பணிகளை மேலாண்மை செய்வதில் கணிப்பொறிகளையும் கணிப்பொறி மென்பொருள் கருவிகளையும் பயன்படுத்துகின்றனர். கணிப்பொறிகள் நமது இல்லங்களிலும் பள்ளிகளிலும், மருத்துவமனைகளிலும், நமது வாழ்வின் அனைத்து நடவடிக்கைகளிலும் நுழைந்துவிட்டன. மக்களின் வாழ்க்கை மற்றும் வேலை முறையையும், நிறுவனங்கள் தமது பணிகளையும் வளங்களையும் மேலாண்மை செய்யும் முறையையும், அரசாங்கங்கள் தமது நாட்டையும் மக்களையும் ஆளும் முறையையும், பல்வேறு முறைமைகளின் நடத்தை மற்றும் செயல்பாட்டு முறையையும் கணிப்பொறிகள் மாற்றி வருகின்றன. வியத்தகு கணிப்பொறித் தொழில்நுட்பங்களின் பயன்பாடுகளுக்கு எல்லையே இல்லை எனலாம்.

பயன்பாட்டுப் பணிகளைப் பரந்த அளவில் கீழ்க்காணுமாறு வகைப்படுத்தலாம்.

- தரவுச் செயலாக்கம் மற்றும் மேலாண்மை (வணிகப் பயன்)
- சொல் செயலாக்கம் (அலுவலக மற்றும் கல்விப் பயன்)
- செய்திப் பரிமாற்றம் (மின்னஞ்சல்)
- படிமச் செயலாக்கம் (பல்லூடகம்)
- எண்முறைக் கணிப்பணி (அறிவியல் பயன்)

நிறுவனங்களில் மேற்கண்ட பணிகளுள் சிலவற்றைச் செய்து முடிக்கத் தேவையான மென்பொருள் கருவிகளைப் பற்றி இந்நூலில் விளக்கப்பட்டுள்ளது. அறிவுக் கூர்மையுடன் இந்த கருவியைப் பயன்படுத்தினால் நிறுவனங்களின் செயல்திறனையும் செயல்வலிமையையும் நிச்சயமாய் மேம்படுத்த முடியும்.

தகவல் செயலாக்கத்துக்கான மென்பொருள் கருவிகள் பல, வெளிச் சந்தையில் கிடைக்கின்றன. இந்நூலில், அமெரிக்காவின் சன் மைக்ரோ சிஸ்டம்ஸ் நிறுவனம் உருவாக்கிய ஸ்டார் ஆஃபீஸ் என்னும் செல்வாக்குப் பெற்ற மென்பொருள் தொகுப்பின் மூலமாகப் பல்வேறு பயன்பாடுகளும் விளக்கப்பட்டுள்ளன. **ஸ்டார் ஆஃபீஸ்** என்பது சொல் செயலாக்கம் (Word Processing), விரிதாள் (Spread Sheet), தரவுத்தளம் (Database), பல்லூடம் (Multimedia) போன்ற கருவிகளின் திரட்டு ஆகும்.

நான் கேட்டேன், மறந்தேன்

நான் பார்த்தேன், மறந்தேன்

நான் செய்தேன், புரிந்துகொண்டேன்

என்னும் அனுபவ மொழியை நான் உறுதியாக நம்புகிறேன்.

செய்முறைப் பயிற்சிகளுக்கு இந்நூல் முழுவதும் முக்கியத்துவம் தரப்பட்டுள்ளது. கற்றல் நேரத்தில் பாதிக்கும் மேலாக மென்பொருள் கருவிகளைச் செயல்படுத்திப் பார்ப்பதற்கே ஒதுக்கப்பட்டுள்ளது.

கணிப்பொறி மற்றும் தகவல் தொழில்நுட்பங்கள் தொடர்ந்து மேம்பட்டு வருகின்றன. உங்களைப் போன்ற இளம் சிந்தனையாளர்கள் வருங்கால உலகை வடிவமைக்கப் போகிறீர்கள். உங்கள் கனவுகள் மற்றும் கடின உழைப்பின்றி, இன்றையச் சிந்தனைகள் நாளைய தொழில்நுட்பங்களாகப் பரிணமிக்கப் போவதில்லை.

"கற்பனை என்பது அறிவைக் காட்டிலும் முக்கியமானது"

–ஆல்பெர்ட் ஈன்ஸ்ட்டீன்

பாட விளக்கங்களைப் பற்பல எடுத்துக்காட்டுகளுடன் செப்பமாகவும், தெளிவாகவும், செய்து பார்க்க எளிமையாகவும் அமைப்பதில் ஆசிரியர் மிகவும் கவனம் செலுத்தியுள்ளார்.

மிகக் குறுகிய காலத்தில் இக்கடினமான பணியைச் செவ்வனே முடித்துத் தந்த இந்நூலின் ஆசிரியர், மீள்பார்வையாளர், பள்ளிக்கல்வி இயக்கக அலுவலர்கள் ஆகியோர் நமது பாராட்டுதலுக்கும், நன்றிக்கும் உரியவர்கள் ஆவர்.

மனிதனின் அன்றாட நடைமுறை வாழ்வில் தீர்க்கப்பட வேண்டிய பிரச்சினைகளை இனிக் கணிப்பொறியின் உதவியின்றித் தீர்க்க முடியாது என்ற நிலைமைக்குத் தள்ளப்பட்டுள்ளோம். நான் கணிப்பொறிகளைக் கண்டு அஞ்சவில்லை. அவை இல்லாவிடில் என்ன ஆகும் என்றே அஞ்சுகிறேன்.

– ஐசக் அசிமோவ்

கணிப்பொறிகள், சிக்கலான அறிவியல், வணிக, நிர்வாகப் பிரச்சினைகளைத் தீர்ப்பதில் நமக்கு உதவுகிற எந்திரங்கள் ஆகும். ஏராளமான தொழிலக, வணிக அமைப்புகளின் தானியங்கமாக்கலுக்கு அவை உதவியுள்ளன. எனினும், அவை மனிதர்களால் உருவாக்கப்பட்டு, மேலாண்மை செய்யப்படும் எந்திரங்கள் என்பதை நாம் நினைவில் கொள்ள வேண்டும்.

அவை தானே சிந்திக்கும் மூளையைப் பெற்றிருக்கவில்லை. அவை செய்கின்ற பணி எதுவாயினும், மனிதரின் ஆணைகளின்படியே செயல்படுகின்றன. நாம் தரும் ஆணைகளை, இருக்கின்ற மென்பொருள்களின் உதவி கொண்டு, நிறைவேற்ற முடியும் என்கிற நிலை இருக்கும் வரை, அவை மிகப் பணிவாக நமது ஆணைகளை நிறைவேற்றுகின்றன. அவை சரியானவையா, பிழையானவையா என்றுகூடப் பார்ப்பதில்லை. அதாவது, கணிப்பொறிகளுக்குத் தன்னறிவு கிடையாது.

எதைச் செய்ய வேண்டும், எப்படி செய்ய வேண்டும், எப்போது செய்ய வேண்டும் என்பதைக் கணிப்பொறிகளுக்குத் தெளிவாக ஆணையிட வேண்டும். கணிப்பொறிகளுக்கு அதுபோன்ற ஆணைகளைத் தருவதற்கான ஒரு வழிமுறையே **நிரலாக்கம்** எனப்படுகிறது. அத்தகைய ஆணைகளை உருவாக்கவும், பரிமாறிக்கொள்ளவும் பயன்படுத்தப்படும் மொழியே **நிரலாக்க மொழி** என அறியப்படுகிறது.

இருநூற்றுக்கு மேற்பட்ட நிரலாக்க மொழிகள் தற்போது பயன்பாட்டில் உள்ளன. அவற்றுள் சில அறிவியல் பயன்பாட்டுக்கு எனவும், சில வணிகப் பயன்பாடுகளுக்கு எனவும் வடிவமைக்கப்பட்டுள்ளன. வேறு சில, பொதுப்பயன் மொழிகள் ஆகும். ஒரு நிரலாக்க மொழி, நிரலர்கள், தீர்வுப் படிநிலைகளை எளிதாக உருவாக்குவதற்கும் வடிவமைப்பதற்கும் உரிய பண்புக்கூறுகளைக் கொண்டிருக்கும்.

நாம் ஏற்கனவே சி–மொழியைக் கற்றுள்ளோம். அது, **செயல்முறை நோக்கு** மொழியாகும். பெயருக்கு ஏற்றுவாறு, தீர்வுக்கான செயல்முறைகளே முக்கியத்துவம் பெற்றன. சி–மொழி சக்திவாய்ந்த பொதுப்பயன் மொழியாகும். இந்நூல் தொகுதி சி – மொழியின் மேம்பட்ட வடிவமான சி++ மொழி பற்றி விளக்குகிறது. சி++, **பொருள்நோக்கு நிரலாக்கம்** (Object Oriented Programming - OOP) என்னும் முற்றிலும் புதிய கருத்துருவைக் கொண்டுள்ளது. எனவே, இம்மொழியை ஒரு பொருள்நோக்கு நிரலாக்கத் தொழில்நுட்பம் என வகைப்படுத்துகின்றனர். நாம் சி++ மொழியைக் கற்பதற்குக் காரணம், இன்றைக்கு அதுவே தொழிலகத் தரப்பாட்டு 'ஊப்' (OOP) மொழியாக ஆகிவிட்டது.

சி++ போன்ற ஊப் மொழிகளில் **பொருள்கள்** (Object) என அறியப்படும் நடப்புலகின் உள்பொருள்களுக்கே முக்கியத்துவம் தரப்படுகிறது. இந்தப் பொருள்கள், நிரல்களில் உடுத்தாளப்பட வேண்டிய ஒரு நபர், ஒரு கார், ஒரு தரவு அட்டவணை அல்லது இது போன்ற எந்தப் பொருளாகவும் இருக்கலாம். மனிதர்களாகிய நாம், அன்றாட வாழ்க்கைப் பிரச்சனைகளை தனித்த பொருள்களின் ஒரு தொகுதியாகவே நோக்குகிறோம். அப்பொருள்களுக்கிடையே நிலவும் உறவுமுறைகளைக் கணக்கில் எடுத்துக்கொண்டு அப்பிரச்சினைகளுக்குத் தீர்வுகாண முயல்கிறோம். அதுபோலவே சி++ மொழியில், நிரலாக்கச் சிக்கல்கள், பொருள்கள் மற்றும் அவற்றுக்கிடையேயான தகவல் பரிமாற்றம் ஆகியவற்றின் அடிப்படையிலேயே பகுத்தாயப்படுகின்றன.

கற்றுணர்வதற்கு எளிய, ஏராளமான எடுத்துக்காட்டுகளும் விளக்க நிரல்களும் இந்நூல் தொகுதியில் வழங்கப்பட்டுள்ளன. சி++ மற்றும் பொருள் நோக்கு நிரலாக்க உலகில் மிக விரைவில் அனுபவம் பெற இந்நூலின் பாடப் பகுதிகள் மாணவர்களுக்கு உதவும் என்பதில் ஐயமில்லை.

தகவல் தொழில்நுட்பம் சார்ந்த பயன்பாடுகள் பலவற்றையும் இந்நூல் தொகுதி விளக்குகிறது. அவற்றைப் புரிந்துகொள்வதற்கு உதவியாக அசைவூட்ட உள்ளடக்க வடிவில் படங்களும் வழங்கப்பட்டுள்ளன (தனியாக ஒரு குறுவட்டில்). அனைத்துப் பயன்பாடுகளிலும் தகவல் தொழில்நுட்பத்தை நன்னெறியுடன் பயன்படுத்துவது பற்றியும் விளக்கப்பட்டுள்ளது.

நூலாசிரியர்கள், மீள்பார்வையாளர்கள், பள்ளிக் கல்வி இயக்கக அலுவலர்கள் அனைவர்க்கும் அவர்களின் மிகச்சிறந்த பணிக்காகவும் ஒத்துழைப்புக்காகவும் என்னுடைய மெய்யான பாராட்டையும் நன்றியையும் தெரிவிக்கக் கடமைப்பட்டுள்ளேன்.

இ. பாலகுருசாமி

தலைவர்

பாடத்திட்ட மீள்பார்வைக் குழு.

பொருளடக்கம்

அதிகாரம் 1	ஸ்டார் ஆஃபிஸ் ரைட்டர் (StarOffice Writer)	பக்கம்
1.1	ஓர் அறிமுகம்	1
1.2	ஒரு புதிய ஆவணத்தை உருவாக்குதல்	2
1.3	ஆவணத்தில் உரையைத் தட்டச்சு செய்தல்	3
1.4	ஆவணங்களைச் சேமித்தல், மூடுதல் மற்றும் திறத்தல்	3
1.5	ஆவணத்தின் பல பகுதிகளுக்குச் செல்லல்	5
1.6	ஆவணத்தை திரை உருளல் செய்தல்	7
1.7	பிழை திருத்துதல்	7
1.8	உரையைச் சேர்த்தல்	7
1.9	உரையைத் தேர்ந்தெடுத்தல்	8
1.10	உரையை நகர்த்துதல்	9
1.11	உரை நகல் எடுத்தல்	10
1.12	வேண்டிய சொல் அல்லது உரைப் பகுதியைக் கண்டுபிடித்து மாற்றுதல்	10
	சுருக்கம்	12
	பயிற்சிகள்	12
அதிகாரம் 2	**உரை வடிவூட்டல்**	
2.1	வடிவூட்டல் விருப்பங்கள்	14
2.2	பத்தி இசைவு	16
2.3	உரையை உள்தள்ளல்	18
2.4	வரி இடைவெளியை மாற்றுதல்	21
2.5	புல்லட் குறி மற்றும் எண்வரிசையை உருவாக்குதல்	21
2.6	ஸ்டைல்களைப் பயன்படுத்தி வடிவூட்டல்	24
2.7	ஸ்டார் ஆஃபிஸ் உதவி	24
	சுருக்கம்	26
	பயிற்சிகள்	27

அதிகாரம் 3 எழுத்துப் பிழை திருத்துதல்

3.1	எழுத்துப் பிழை திருத்துதல்	28
3.2	ஆவணத்தை தட்டச்சு செய்து முடித்தபின் எழுத்துப்பிழை சரி செய்தல்	28
3.3	தானியங்கு சரி செய்யும் தேர்வு	30
3.4	தானியங்கு சரி செய்யும் பட்டியலில் ஒரு சொல்லைச் சேர்த்தல்	31
	சுருக்கம்	32
	பயிற்சிகள்	32

அதிகாரம் 4 அட்டவணையில் வேலை செய்தல்

4.1	ஒரு எளிய அட்டவணையைச் செய்தல்	34
4.2	அட்டவணையில் தரவுகளை நுழைத்தல்	35
4.3	வரிசை மற்றும் நெடுவரிசைகளை நீக்குதல் மற்றும் சேர்த்தல்	35
4.4	குறுக்கு வரிசை / நெடுவரிசையின் அகலத்தை மாற்றுதல்	36
4.5	அட்டவணை வடிவூட்டல் கருவிப்பட்டை	37
	சுருக்கம்	39
	பயிற்சிகள்	39

அதிகாரம் 5 பக்க வடிவமைப்பு

5.1	ஓரத்தை மாற்றுதல்	41
5.2	பக்கத்தின் அமைவுகள்	42
5.3	தலைப்பு அடிக்குறிப்புகளை உருவாக்குதல்	42
	சுருக்கம்	45
	பயிற்சிகள்	45

அதிகாரம் 6 அட்டவணைச் செயலி

6.1	முன்னுரை	47
6.2	ஸ்டார் ஆஃபிஸின் ஸ்டார் கால்க் பயன்படுத்துவது எப்படி ?	51
6.3	அட்டவணைக் கோப்பின் தரவுகளை பதிப்பாய்வு செய்தல்	56
6.4	வாய்ப்பாடுகளை அட்டவணையில் உள்ளிடல்	57
6.5	FILL கட்டளை	59

6.6	நுண்ணறைப் பார்வையிடல்	62
6.7	சார்புகளின் பயன்பாடு	62
6.8	தேதிக் கணக்கீடு	65
6.9	அட்டவணைத் தாளை வடிவமைத்தல்	66
6.10	அட்டவணைத் தாளின் நெடுவரிசை அகலத்தையும் வரிசையின் உயரத்தையும் மாற்றுதல்	68
6.11	அட்டவணைத் தாளில் நுண்ணறைகள், வரிசைகள் நெடுவரிசைகள் ஆகியவற்றைச் சேர்த்தல்	70
6.12	அட்டவணைத் தாளிலிருந்து நுண்ணறைகள், வரிசைகள், நெடுவரிசைகள் இவற்றை நீக்குதல்	72
6.13	அட்டவணைத்தாளில் படங்களையும் சிறப்புக் குறியீடுகளையும் சேர்த்தல்	73
6.14	அட்டவணைத்தாளில் வரைபடம் வரைதல்	74
6.15	பொருள் சேர்த்தல்	75
6.16	அட்டவணைத் தாளில் வரைபடங்களுடன் செயல்படல்	76
6.17	ஒன்றுக்கு மேற்பட்ட அட்டவணைக் கோப்புகளில் செயலாற்றல்	79
6.18	அட்டவணைத் தாள்களை அச்சிடல்	80
6.19	Star Calc–இன் தரவுத்தள சார்புகள்	81
	சுருக்கம்	81
	பயிற்சிகள்	82

அதிகாரம் 7 தரவுத்தளம்

7.1	முன்னுரை	84
7.2	தரவும் தகவலும்	84
7.3	தரவுச் செயலாக்கம்	86
7.4	தரவுத் தளம்	89
7.5	தரவுத்தள மேலாண்மையின் அடிப்படைத் தத்துவங்கள்	94
7.6	StarBase–இல் செயலாற்றல்	94
7.7	அலுவலகப்பணி தானியக்க பயன்பாடுகளுடன் ஒருங்கிணைத்தல்	119
	சுருக்கம்	127
	பயிற்சிகள்	128

அதிகாரம் 8 பல்லூடக அறிமுகம்

8.1	பல்லூடகம் என்றால் என்ன ?	130
8.2	பல்லூடகப் பயன்பாடுகள்	130
8.3	பல்லூடகக் கூறுகள் : நிழற்படங்கள், ஒலி, அசைவுப்படங்கள் & ஒளிக்காட்சி	131
8.4	பல்லூடகப் பயன்பாடு	138
	சுருக்கம்	139
	பயிற்சிகள்	139

அதிகாரம் 9 நிகழ்த்துதல்

9.1	அறிமுகம்	140
9.2	ஓர் அடிப்படை நிகழ்த்தல்	140
9.3	நிகழ்த்தலைக் கையாளல்	147
9.4	நிகழ்த்தலை தனிப்பயனாக்கல்	160
9.5	நிகழ்த்தலை அச்சிடல்	164
	சுருக்கம்	173
	பயிற்சிகள்	174
	இணைப்பு : மேலும் படிக்க	175

அதிகாரம் 10 சி++ மொழியின்வழி பொருள்நோக்கு நிரலாக்கக் கருத்துருக்கள் (Object Oriented Concepts using C++)

10.1	பொருள்நோக்கு நிரலாக்க மொழி **(Object Oriented Programming Language)**	176
10.2	பல்லுருவாக்கம் **(Polymorphism)**	178
10.3	மரபுரிமம் **(Inheritance)**	179
10.4	ஒரு நடைமுறை எடுத்துக்காட்டு : வீட்டு நீர்ப்பயன்பாடு	181
	பயிற்சி வினாக்கள்	183

அதிகாரம் 11 சி++ மொழியின் முன்னோட்டம் (Overview of C++)

11.1	முன்னுரை	185
11.2	சி++ குறியுருத் தொகுதி **(C++ Character Set)**	185
11.3	தரவு இனங்கள் **(Data types)**	202
11.4	மாறிகள் **(Variables)**	210
	பயிற்சி வினாக்கள்	217

அதிகாரம் 12 அடிப்படைக் கூற்றுகள் (Basic Statements)

12.1	உள்ளீட்டு / வெளியீட்டுக் கூற்றுகள்	220
12.2	முதல் சி++ நிரல் – சி++ நிரலின் கட்டமைப்பு	222
12.3	அழைப்புக் கூற்றுகள் **(Declaration Statements)**	222
12.4	மதிப்பிருத்து கூற்றுகள் **(Assignment Statements)**	223
12.5	கட்டுப்பாட்டுக் கட்டளை அமைப்புகள் **(Control Structures)**	223
12.6	நிரல் உருவாக்கம்	224
	பயிற்சி வினாக்கள்	251

அதிகாரம் 13 செயற்கூறுகள் (Functions)

13.1	முன்னுரை	256
13.2	செயல்கூறின் முன்வடிவு **(Function Prototype)**	257
13.3	ஒரு செயற்கூறினை அழைத்தல்	259
13.4	செயற்கூறினுக்கு அளபுருக்களை அனுப்பிவைத்தல்	259
13.5	மதிப்பினைத் திருப்பியனுப்புதல்	268
13.6	inline செயற்கூறுகள்	271
13.7	மாறிகளின் வரையெல்லை **(scope)** விதிமுறைகள்	272
	பயிற்சி வினாக்கள்	276

அதிகாரம் 14 கட்டமைப்புத் தரவினம் – அணிகள் (Structured Data Type - Arrays)

14.1 முன்னுரை 282

14.2 ஒருபரிமாண அணி **(Single Dimensional Array)** 283

14.3 சரங்கள் **(Strings)** 288

14.4 இருபரிமாண அணி **(Two-Dimensional Array)** 293

14.5 சரங்களின் அணி 299

பயிற்சி வினாக்கள் 301

அதிகாரம் 15 இனக்குழுக்களும் பொருள்களும் (Classes and Objects)

15.1 இனக்குழுக்கள் – ஒரு முன்னுரை 306

15.2 இனக்குழுவை வரையறுத்தல் 306

15.3 தரவு அருவமாக்கம் **(Data Abstraction)** 308

15.4 தரவு உறுப்புகளும் உறுப்புச் செயற்கூறுகளும் 309

15.5 பொருள்களை உருவாக்குதல் 310

15.6 இனக்குழு உறுப்புகளை அணுகுதல் 310

15.7 இனக்குழுவின் செயற்கூறுகளை வரையறுத்தல் 312

15.8 பொருள்களுக்கு நினைவக ஒதுக்கீடு 313

15.9 static தரவு உறுப்புகள் 314

15.10 பொருள்களின் அணிகள் 317

பயிற்சி வினாக்கள் 318

அதிகாரம் 16 பல்லுருவாக்கம் (Polymorphism)

16.1 முன்னுரை 322

16.2 செயற்கூறு பணிமிகுப்பு **(Function Overloading)** 322

16.3 செயற்குறிப் பணிமிகுப்பு **(Operator Overloading)** 326

பயிற்சி வினாக்கள் 334

அதிகாரம் 17 ஆக்கிகளும் அழிப்பிகளும் (Constructors and Destructors)

17.1 முன்னுரை 335
17.2 ஆக்கிகள் **(Constructors)** 335
17.3 ஆக்கியின் செயல்பாடுகள் 336
17.4 ஆக்கியின் பணிமிகுப்பு **(Constructor Overloading)** 336
17.5 ஆக்கி வரையறுப்பு மற்றும் பயன்படுத்தலுக்கான விதிமுறைகள் 341
17.6 அழிப்பிகள் **(Destructors)** 341
17.7 அழிப்பி வரையறுப்பு மற்றும் பயன்படுத்தலுக்கான விதிமுறைகள் 342
பயிற்சி வினாக்கள் 343

அதிகாரம் 18 மரபுரிமம் (Inheritance)

18.1 முன்னுரை 344
18.2 மரபுரிமத்தின் பலன்கள் 345
18.3 தருவிக்கப்பட்ட இனக்குழுவும் அடிப்படை இனக்குழுவும் 345
18.4 காண்புநிலைப் பாங்கு / அணுகியல்பு வரையறுப்பி **(Visibility Mode / Accessibility Specifier)** 348
18.5 மரபுரிமத்தின் வகைகள் 351
பயிற்சி வினாக்கள் 354

அதிகாரம் 19 சமுதாயத்தின் மீது கணிப்பொறியின் தாக்கம் (Impact of Computers on Society)

19.1 முன்னுரை 358
19.2 சொந்தப் பயன்பாட்டுக்குக் கணிப்பொறிகள் 358
19.3 கணிப்பொறி மயமாக்கப்பட்ட வீடுகள் 359
19.4 வீட்டிலிருந்தே வங்கிப் பணியும் கடைப்பொருள் வாங்கலும் 360
19.5 கல்வியில் கணிப்பொறிகள் 362
19.6 பொழுதுபோக்கில் கணிப்பொறிகள் 363
19.7 நலவாழ்வுக்குக் கணிப்பொறிகள் 365

19.8 வேளாண்மைத் துறையில் கணிப்பொறிகள் 366

19.9 நிகழ்நேரப் பயன்பாடுகளில் இணையம் 367

பயிற்சி வினாக்கள் 367

அதிகாரம் 20 தகவல் தொழில்நுட்பம் சார்ந்த சேவைகள் (IT Enabled Services)

20.1 முன்னுரை 368

20.2 மின்– அரசாண்மை **(e-Governance)** 369

20.3 அழைப்புதவி மையங்கள் **(Call-Centres)** 370

20.4 தரவு மேலாண்மை **(Data Management)** 370

20.5 மருத்துவப் பெயர்ப்பாவணமும் தொலைமருத்துவமும் **(Medical Transcription and Tele-Medicine)** 372

20.6 கணிப்பொறித் தரவாக்கம்**(Data Digitization)** 373

20.7 வலையகச் சேவைகள் 374

பயிற்சி வினாக்கள் 374

அதிகாரம் 21 கணிப்பொறி நன்னெறி (Computer Ethics)

21.1 தரவுப் பாதுகாப்பு **(Data Security)** 376

21.2 கணிப்பொறிக் குற்றம் **(Computer Crime)** 376

21.3 அரண் உடைத்தல் **(Cracking)** 378

21.4 வேலை, குடும்பம், பொழுதுபோக்கு 378

பயிற்சி வினாக்கள் 378

அதிகாரம் 1

ஸ்டார் ஆஃபிஸ் ரைட்டர் (Star Office Writer) ஓர் அறிமுகம்

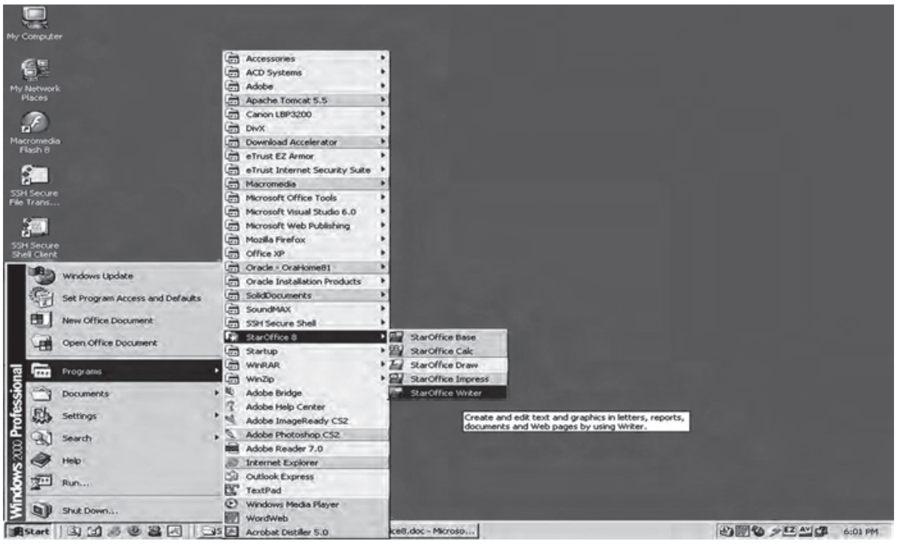
1.1 முன்னுரை

ஸ்டார் ஆஃபிஸ் பல பயன்பாடுகளைக் (applications) கொண்ட ஒரு தொகுப்பாகும். இந்தப் பயன்பாடுகள் அனைத்தும் ஒன்றாகச் சேர்ந்து ஒரு ஒருங்கிணைந்த சூழலாக (integrated environment) உள்ளது. ஸ்டார் ஆஃபிஸ் பயன்பாடுகளில் ஒரு சில கீழ்க்கண்ட பட்டியலில் தரப்பட்டுள்ளன.

1. உரை ஆவணங்களை (text document) உருவமைக்கப் பயன்படும் ஸ்டார் ஆஃபிஸ் ரைட்டர்(StarOffice Writer).
2. அட்டவணைச்செயலி (SpreadSheets) உருவமைக்கப் பயன்படும் ஸ்டார் ஆஃபிஸ் கால்க் (StarOffice Calc).
3. நிகழ்த்துதல் (presentation) உருவமைக்கப் பயன்படும் ஸ்டார் ஆஃபிஸ் இம்ப்ரெஸ் (StarOffice Impress).
4. படங்கள் வரைவதற்கு உதவியாக இருக்கும் ஸ்டார் ஆஃபிஸ் ட்ரா (StarOffice Draw).
5. தரவுத் தளம் (Database) உருவாக்கப் பயன்படும் ஸ்டார் ஆஃபிஸ் பேஸ் (StarOffice Base).

ஸ்டார் ஆஃபிஸ்ஸைத் துவங்குவதற்கு கீழே குறிப்பிட்டுள்ளவாறு செய்க:

1. பணிப்படையில் உள்ள Start பொத்தானை அழுத்தவும். பிறகு தோன்றும் இணைப்புகளில் Program பொத்தானை அழுத்தவும்.
2. தோன்றும் மென்பொருள் தொகுப்புகளிலிருந்து StarOffice8 தொகுப்பைத் தேர்வு செய்க.
3. பிறகு தோன்றும் StarOffice பயன்பாடுகளில் உங்களுக்கு வேண்டிய பயன்பாட்டின் மீது கிளிக் செய்யவும். (படம் 1.1)



படம் 1.1 ஸ்டார் ஆபிஸ் – 8 – இன் பயன்பாடுகள்

StarOffice- இன் பிற பயன்பாடுகளை StarOffice Writer பயன்பாட்டிலிருக்கும் போது திறக்க, File → New பொத்தான்களை அழுத்தி, தேவையான பயன்பாட்டின் மீது கிளிக் செய்க.

விண்டோஸ், யூனிக்ஸ், போன்ற பல்வேறு வகையான இயக்க அமைப்புகளிலும் வேலை செய்யும் வகையில் வடிவமைக்கப்பட்ட ஒரு பயன்பாடுதான் ஸ்டார் ஆஃபிஸ் ஆகும். பல்வேறு இயக்க அமைப்புகளுக்கும் பொதுவான ஒரு சூழலைக் கொடுக்கும் விதத்தில் ஸ்டார் ஆஃபிஸ் சொந்தமாக ஒரு டெஸ்க்டாப்பைப்(desktop)பயன்படுத்துகிறது. இதற்கு ஸ்டார் டெஸ்க்டாப் (Star desktop) என்று பெயர். இந்த டெஸ்க்டாப் மைக்ரோ ஸாஃப்ட் விண்டோஸ்களின் டெஸ்க்டாப் போன்றே இருக்கும். ஸ்டார்ட்(Start) பொத்தானைக் கொண்டுள்ள ஒரு பணிப்பட்டை மற்றும் விண்டோஸ் டெஸ்க்டாப்பில் உள்ள அனைத்துப் பணிக்குறிகளையும் கொண்டதாக இந்த டெஸ்க்டாப் இருக்கும். அவற்றை விண்டோஸ் சூழலில் பயன்படுத்துவதைப் போலவே பயன்படுத்த முடியும்.

1.2 ஒரு புதிய ஆவணத்தை உருவாக்குதல்

ஸ்டார் ஆஃபிஸ் ரைட்டர் ஒரு சொற்செயலி (Word Processor) ஆகும். கணிப்பொறியில் பொருத்தமான ஒரு மென் பொருளைப் பயன்படுத்தி ஒரு ஆவணத்தை உருவாக்குதல், பார்த்தல், திருத்துதல், சேமித்தல், திரும்ப எடுத்தல் (retrieving) மற்றும் அச்சிடுதல் ஆகிய செயற்பாடுகளைச் செய்வதை சொற்செயலாக்கல் என்று அழைக்கிறோம். ஒரு ஆவணத்தில் உரைகள், பட்டியல்கள், வரைபடங்கள், படங்கள், சமன்பாடுகள் முதலியன இருக்கலாம்.

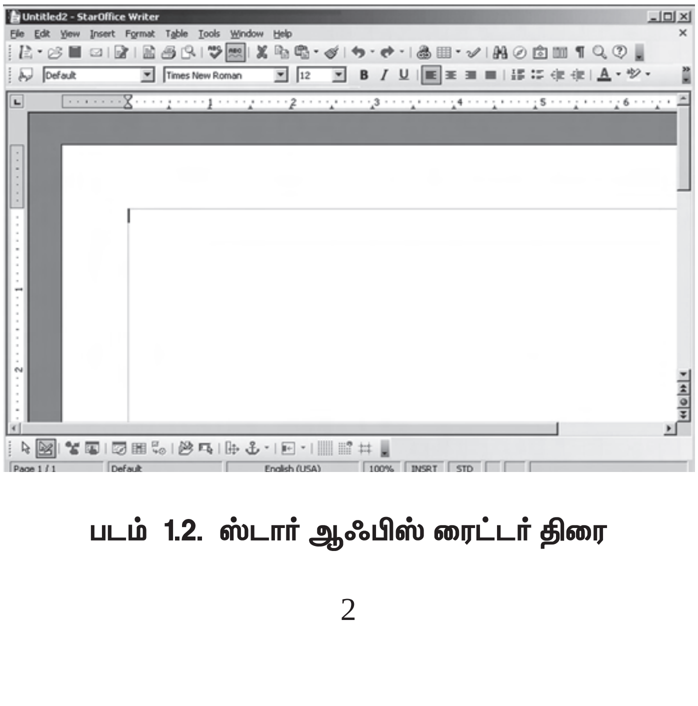
M.S. Word, Lotus Amipro. Word star. Word pro ஆகியவை மென்பொருள் சந்தையில் கிடைக்கும் இதர சொற்செயலிகள் (Word Processors) ஆகும்.

ஸ்டார் ஆஃபிஸ் ரைட்டரில் ஒரு புதிய ஆவணத்தைக் கீழ்க்கண்டவாறு திறக்கலாம்.

1. ஸ்டார் ஆபிஸ் (StarOffice)சூழலில் (environment) Start பட்டியைக் கிளிக் செய்ய வேண்டும்.

2. உரை ஆவணம் (Text Document) என்ற குறும்படத்தைக் கிளிக் செய்ய வேண்டும். இப்பொழுது Untitled என்ற பெயரில் ஒரு புதிய ஆவணம் திறக்கப்படும்.

இதற்கு மாறாக, File → New →Text Document என்ற கட்டளைகளை உபயோகித்தும் ஒரு புதிய ஆவணத்தைத் திறக்கலாம்.



படம் 1.2. ஸ்டார் ஆஃபிஸ் ரைட்டர் திரை

ஸ்டார் ஆஃபிஸ் ரைட்டரில் ஒரு புதிய ஆவணத்தைத் திறந்தவுடன் திரை படம் 1.2 ல் காட்டியவாறு தோன்றும்.

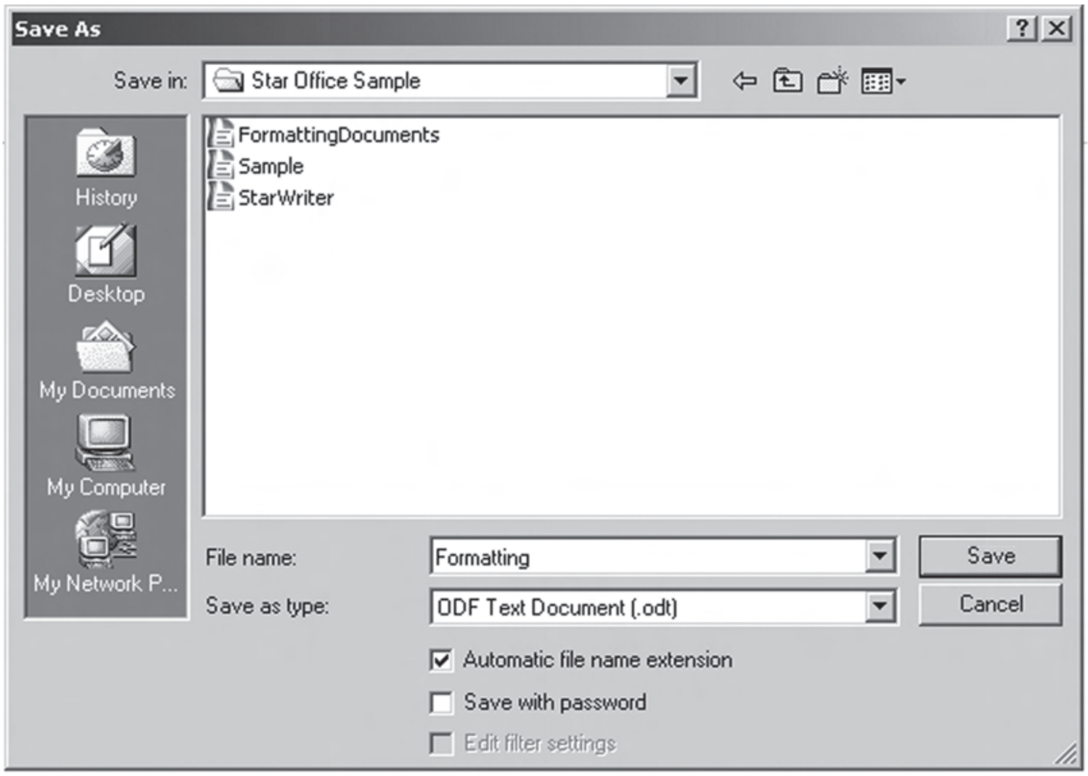
1.3 ஆவணத்தில் உரையைத் தட்டச்சு செய்தல்

புதிதாக ஆவணம் திறக்கப்பட்டவுடன், திரையில் உள்ள பெரிய காலியான இடத்தில்தான் ஸ்டார் ஆஃபிஸ் ரைட்டர் ஆவணத்தைத் தட்டச்சு செய்ய வேண்டும். ஒவ்வொரு எழுத்தாகத் தட்டச்சு செய்யும் பொழுது விட்டு விட்டுத் தோன்றும் (flashing) நெடும்பட்டை (Vertical bar) ஒன்று வலது புறமாக நகர்ந்து கொண்டே இருக்கும், இதற்கு செருகும் இடம் (Insertion Point) என்று பெயர். புதிதாகத் தட்டச்சு செய்யப்படும் உரை இந்த இடத்தில்தான் தோன்றும். ஒரு வரியின் கடைசி வரை தட்டச்சு செய்து விட்டால் அதைத் தொடர்ந்து வரும் எழுத்துக்கள்தானாகவே மடிந்து (wrap up) அடுத்த வரிக்கு வந்து விடும். நுழைவுப் (Enter)பொத்தானை ஒவ்வொரு வரியின் இறுதியிலும் அழுத்த வேண்டியது இல்லை. ஒரு பத்தியின் கடைசியிலோ அல்லது ஒரு காலிவரி வேண்டுமென்றாலோ மட்டும் தான் நுழைவுப் பொத்தானைப் பயன்படுத்த வேண்டும். ஒரு பக்கம் நிறைந்தவுடன் தானாகவே அடுத்த பக்கம் உருவாக்கப்பட்டு விடும்.

1.4 ஆவணங்களைச் சேமித்தல், மூடுதல் மற்றும் திறத்தல்

1.4.1 ஆவணத்தைச் சேமித்தல்

முதன்முறையாக ஒரு ஆவணத்தைச் சேமிக்கும் பொழுது, ஸ்டார் ஆஃபிஸ் ரைட்டர் அந்த ஆவணத்தை எந்தப் பெயரில் சேமிப்பது என்று பயனரை வினவும். ஆவணத்திற்கு ஒரு பெயரிடுவதன் மூலம் அதனைப் பிறிதொரு நேரத்தில் தேடுவதும், திறப்பதும் இயலும், மேலும், எந்த இயக்கி (drive) மற்றும் எந்த உறையில் (folder) அந்த ஆவணத்தைச் சேமிக்க வேண்டும் என்பதையும் சொல்ல வேண்டும்.



படம் 1.3 சேமிப்பு உரையாடல் பெட்டி

முதன் முறையாக ஒரு ஆவணத்தை சேமிப்பதற்கான வழிகள் பின்வருமாறு :

1. File→Save என்ற கட்டளையையோ அல்லது என்ற குறும் படத்தையோ கிளிக் செய்ய வேண்டும். படம் 1.3–இல் காட்டியவாறு ஒரு சேமிப்பு உரையாடல் பெட்டி (Save As dialog box) தோன்றும்.

2. ஒரு குறிப்பிட்ட இயக்கியைத் (drive) தேர்வு செய்ய என்ற பணிக் குறியை இருமுறை கிளிக் செய்ய வேண்டும்.

3. பின் இயக்கியின் மீது இருமுறை கிளிக் செய்ய வேண்டும். இப்பொழுது அந்த இயக்கியில் உள்ள உறைகளின் (folders) பெயர்கள் திரையிடப்படும்.

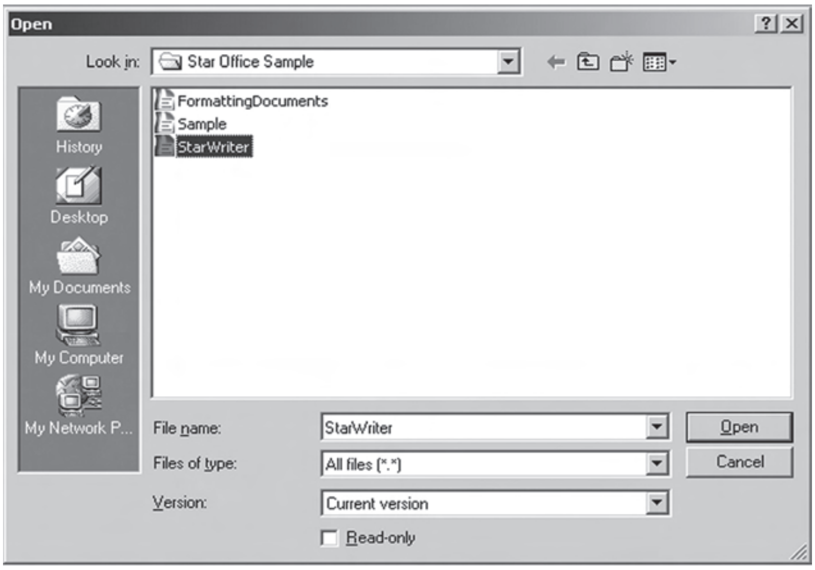
4. தேவையான உறையின் மீது இருமுறை கிளிக் செய்து கோப்புப் பெயர் பட்டியல் பெட்டியில் (file name list box) ஆவணத்துக்குப் பெயர் கொடுக்க வேண்டும்.

5. எந்த வகை ஆவணமாகச் சேமிக்க வேண்டும் என்பதை கோப்பு வகைப் பட்டியல் பெட்டி (file type list box) யில் தட்டச்சு செய்து **Save** பொத்தானைக் கிளிக் செய்ய வேண்டும். இப்பொழுது ஆவணம் சேமிக்கப்பட்டு அந்த ஆவணத்தின் பெயர் தலைப்புப் பட்டியில் (title bar) தோன்றும்.

ஒருமுறை ஆவணம் சேமிக்கப்பட்டபின் மீண்டும் அதனைச் சேமிப்பதற்கு பெயரை மீண்டும் தர வேண்டியது இல்லை. File → Save என்ற கட்டளையின் மூலம் சேமித்துக் கொள்ளலாம். அல்லது Ctrl + S என்ற விசைப்பலகை குறுக்கு வழியைப் (keyboard Shortcut) பயன்படுத்தலாம்.

1.4.2 ஆவணத்தை மூடுதல்

ஒரு ஆவணத்தைச் சேமித்தவுடன் அது தானாகவே மூடப் படுவதில்லை. பயனர் தொடர்ந்து அந்த ஆவணத்தில் வேலை செய்யும் வகையில் அது திறந்தே இருக்கும். வேலை முடிந்தவுடன், ஆவணத்தைச் சேமித்து மூடி விட வேண்டும். இதற்கு File→Close என்ற கட்டளையைப் பயன்படுத்தலாம்.



படம் 1.4 உரையாடல் பெட்டி

1.4.3 ஆவணத்தை திறத்தல்

சேமிக்கப்பட்டு, மூடப்பட்ட ஒரு ஆவணத்தை மீண்டும் திறப்பதற்கு File→Open என்ற கட்டளையைப் பயன்படுத்த வேண்டும். இப்பொழுது படம் 1.4–இல் காட்டியவாறு ஒரு உரையாடல் பெட்டி தோன்றும். இந்த உரையாடல் பெட்டி Save As உரையாடல் பெட்டியைப் போன்றதே ஆகும். திறக்கப்பட வேண்டிய கோப்பின் பெயரை திரையிடப்பட்டுள்ள பட்டியலிலிருந்து தேர்வு செய்ய வேண்டும்.

தேவையான கோப்பைத் திறப்பதற்கான மாற்றுவழி, என்ற பணிக் குறியைக் கிளிக் செய்வது ஆகும்.

1.4.4 ஒன்றுக்கு மேற்பட்ட ஆவணங்களில் வேலை செய்தல்

ஸ்டார் ஆஃபிஸ் ரைட்டரில் வேலை செய்யும் பொழுது ஒன்றுக்கு மேற்பட்ட ஆவணங்களை ஒரே நேரத்தில் திறந்து வைத்துக் கொள்ள முடியும். வேண்டிய ஆவணங்களை ஒன்றன் பின் ஒன்றாகத் திறக்க வேண்டும். இதன்பின் ஒரு ஆவணத்திலிருந்து இன்னொரு ஆவணத்துக்கு மாறுவதற்கு கீழ்க்கண்ட இருவழிகள் உள்ளன.

1. பட்டிப் பட்டையில் Window என்ற தேர்வைக் கிளிக் செய்து, பின்னர் கிடைக்கும் கோப்புப் பெயர் பட்டியலில் வேண்டிய கோப்பின் மீது கிளிக் செய்ய வேண்டும்.

2. பணிப்பட்டையில் (task bar) தோன்றும் ஆவணத்தின் பெயரையும் கிளிக் செய்யலாம். இந்த ஆவணங்களை ஒன்றன்பின் ஒன்றாக மூடுவதற்கு File → Close என்ற கட்டளையைப் பயன்படுத்தலாம்.

1.5 ஆவணத்தின் பல பகுதிகளுக்குச் செல்ல

ஆவணத்தின் எந்த ஒரு இடத்திற்கும் செருகும் புள்ளியை நகர்த்த சுட்டெலி (Mouse)அல்லது விசைப் பலகையை உபயோகிக்கலாம். பக்கத்தின் இறுதியில் தோன்றும் ஒரு பட்டையான படுக்கைக்கோடு ஆவணத்தின் இறுதியைக் குறிக்கும். செருகும் புள்ளியை இதற்குப் பின்னால் நகர்த்த முடியாது.

சுட்டெலி மூலம் செருகும் புள்ளியை நகர்த்துவதற்கு சுட்டியை வேண்டிய இடத்துக்கு நகர்த்தி ஒரு கிளிக் செய்ய வேண்டும். சுட்டெலியின் சுட்டியும், செருகும் புள்ளியும் வெவ்வேறானவையாகும்.

விசைப்பலகையின் மூலம் செருகும் புள்ளியை நகர்த்துவதற்கு அம்புக் குறியையோ அல்லது பொத்தான்களையோ பயன்படுத்த வேண்டும். இதற்கெனப் பயன்படும் பொத்தான்கள், பட்டியல் 1.1 ல் கொடுக்கப்பட்டுள்ளன. சாவிச் சேர்மானத்தில் (key combination) உள்ள கூட்டல் குறி முதல் பொத்தானை அழுத்திப் பிடித்தவாறு இரண்டாவது பொத்தானை அழுத்த வேண்டும் என்பதைக் குறிக்கிறது.

பட்டியல் 1.1. சாவிப்பலகையிலுள்ள அசைவு (movement) ப் பொத்தான்கள்

நகர்த்துவதற்கு	அழுத்த வேண்டிய சாவிகள்
1. இடப்புறமாக ஒரு எழுத்து அளவுக்கு	Left arrow
2. வலப்புறமாக ஒரு எழுத்து அளவுக்கு	Right arrow
3. இடப்புறமாக ஒரு சொல் அளவுக்கு	Ctrl+Left arrow
4. வலப்புறமாக ஒரு சொல் அளவுக்கு	Ctrl+Right arrow
5. மேல்புறமாக ஒருவரி அளவுக்கு	Up Arrow
6. கீழ்புறமாக ஒரு வரி அளவுக்கு	Drown Arrow
7. வரியின் இறுதிக்கு	End
8. வரியின் தொடக்கத்திற்கு	Home
9. ஆவணத்தின் இறுதிக்கு	Ctrl+End
10. ஆவணத்தின் தொடக்கத்திற்கு	Ctrl+Home

செய்து பார்

1. கீழ்க்கண்ட உரையைத் தட்டச்சு செய்யவும்

According to legend, the ancient Olympic Games were founded by Heracles. Yet the first Olympic Games for which we still have written records were held in 776 BCE (though it is generally believed that the Games had been going on for many years already). At this Olympic Games, a naked runner, Coroebus (a cook from Elis), won the sole event at the Olympics, the stade - a run of approximately 192 meters (210 yards). This made Coroebus the very first Olympic champion in history.

The ancient Olympic Games grew and continued to be played every four years for nearly 1200 years.

2. மேலே கண்ட உரையைத் தட்டச்சு செய்தபின் அதை Exercise 1 என்ற பெயரில் சேமித்து கோப்பை மூடவும்

3. Exercise 1 என்ற ஆவணத்தைத் திறந்து அதில் கீழ்க்கண்ட பத்தியைச் சேர்க்கவும். பின் கோப்பை சேமித்து மூடவும்.

The very first modern Olympic Games opened in the first week of April 1896. Since the Greek government had been unable to fund construction of a stadium, a wealthy Greek architect, Georgios Averoff, donated one million drachmas (over $100,000) to restore the Panathenaic Stadium, originally built in 330 BCE, with white marble for the Olympic Games.

1.6 ஆவணத்தை திரை உருளல் செய்தல்

ஒரு ஆவணம் நீண்டதாக இருந்தால் செருகும் புள்ளியை நகர்த்தாமலேயே உரையை திரைஉருளல் செய்யமுடியும். இதற்கு சுட்டெலியும், திரை உருளல் அம்புக் குறிகளும் திரை உருளல் பட்டைகளும் உதவும். மேலும் கீழுமாக நகர்த்துவதற்கும், கிடையாக நகர்த்துவதற்கும் என தனித்தனி திரை உருளல் அம்புக் குறிகள் உள்ளன.

திரை உருளல் செய்யும் முறை பின்வருமாறு

1. இட – வலமாக உரையை நகர்த்த இடது மற்றும் வலது அம்புக் குறிகளை கிளிக் செய்ய வேண்டும்.

2. மேலும் கீழுமாக திரை உருளல் செய்ய மேல் மற்றும் கீழ் அம்புக் குறிகளை கிளிக் செய்ய வேண்டும்.

3. ஒரு குறிப்பிட்ட தூரம் திரைஉருளல் செய்ய திரை உருளல் பெட்டியை (Scroll Box) மேலாகவோ கீழாகவோ இழுக்க வேண்டும். ஆவணத்தில் பல பக்கங்கள் இருந்தால் நடப்பு பக்கம் எது என்பதை திரை உருளல் பெட்டிக்கு பக்கத்தில் தோன்றும் மேல் மீட்பு பக்க எண் மூலம் தெரிந்து கொள்ளலாம்.

ஆவணத்தை திரைஉருளல் செய்தால் செருகும் புள்ளி நகராது. அதை நகர்த்துவதற்கு வேண்டிய இடத்தில் சுட்டெலியைக் கிளிக் செய்ய வேண்டும்.

செய்து பார்

1. முன்பு Exercise 1 என்று சேமித்த ஆவணத்தைத் திறந்து அதன் பல பகுதிகளுக்குச் செல்ல, சாவிப்பலகைத் தேர்வுகள் அனைத்தும் வேலை செய்கிறதா என்று பார்க்கவும். மேலும் திரை உருளல் மற்றும் திரை உருளல் அம்புக் குறிகளையும் பயன்படுத்திப் பார்க்கவும்.

1.7 பிழை திருத்தல்

தட்டச்சு செய்யப்படும் அனைத்து எழுத்துக்களும் திரையில் தோன்றும். ஏதாவது தவறு தெரிந்தால் அதனை Backspace பொத்தான் அல்லது Delete பொத்தானைக் கொண்டு திருத்தலாம். Backspace பொத்தான் செருகும் புள்ளிக்கு இடது பக்கத்தில் உள்ள எழுத்துக்களையும் Delete பொத்தான் செருகும் புள்ளிக்கு வலது பக்கத்தில் உள்ள எழுத்துக்களையும் அழிக்கும்.

1.8 உரையைச் சேர்த்தல்

ஆவணம் முழுமையாக தட்டச்சு செய்யப்பட்ட பின் தேவையானால் வேண்டிய உரையை ஆவணத்தின் முதலிலோ இறுதியிலோ அல்லது இடையிலோ சேர்க்க முடியும். வேண்டிய உரையைச் சேர்ப்பதற்கு செருகும் புள்ளியை வேண்டிய இடத்தில் வைத்து விட்டு அதன் பின்னர் தட்டச்சு செய்ய வேண்டும். புதிய உரை இப்பொழுது சேர்க்கப்பட்டு ஏற்கனவே உள்ள உரை வலது பக்கத்தில் நகரும். புதிய உரையை மேல் அச்சிடல் (Type Over) நிலை அல்லது செருகல்

(Insert) நிலை ஆகிய இரு நிலைகளில் சேர்க்கலாம். மேல் அச்சிடல் (Type over) நிலையில் புதிய உரை சேர்க்கப்படும் போது செருகும் புள்ளிக்கு வலது பக்கத்தில் உள்ள எழுத்துக்கள் அழியும். செருகல் நிலையில் புதிய உரை செருகும் புள்ளி உள்ள இடத்தில் இடப்படும். Insert பொத்தானைப் பயன்படுத்தி மேல் அச்சிடல் அல்லது செருகல் நிலை ஆகியவற்றில் ஒன்றிலிருந்து இன்னொன்றுக்கு மாறிக் கொள்ளலாம். நிலைமைப் பட்டையின் மூலம் நடப்பு முறைமை என்ன என்பதைத் தெரிந்து கொள்ளலாம்.

பிழையைத் திருத்தும் போது அவ்வப்போது ஆவணத்தை File → Save கட்டளையின் மூலம் சேமித்துக் கொள்ள வேண்டும்.

1.9 உரையைத் தேர்ந்தெடுத்தல்

ஆவணத்தை உருவாக்கும் போது ஒவ்வொரு எழுத்தாக தட்டச்சு செய்தாலும் திருத்தும் போதும் வடிவூட்டும் போதும் சொற்களையோ, வரிகளையோ, பத்திகளையோ அல்லது சில நேரம் முழு ஆவணத்தையோ தேர்ந்தெடுக்க வேண்டி வரும். உரையை தேர்ந்தெடுத்த பின்னால் தேவையான மாற்றங்களை செய்து கொள்ளலாம். உரையை நகர்த்தவும், நகல் எடுக்கவும், தடிப்பாக்கவும் முடியும். உரையை தேர்ந்தெடுப்பதற்கு சுட்டெலி அல்லது விசைப் பலகையை பயன்படுத்தலாம்.

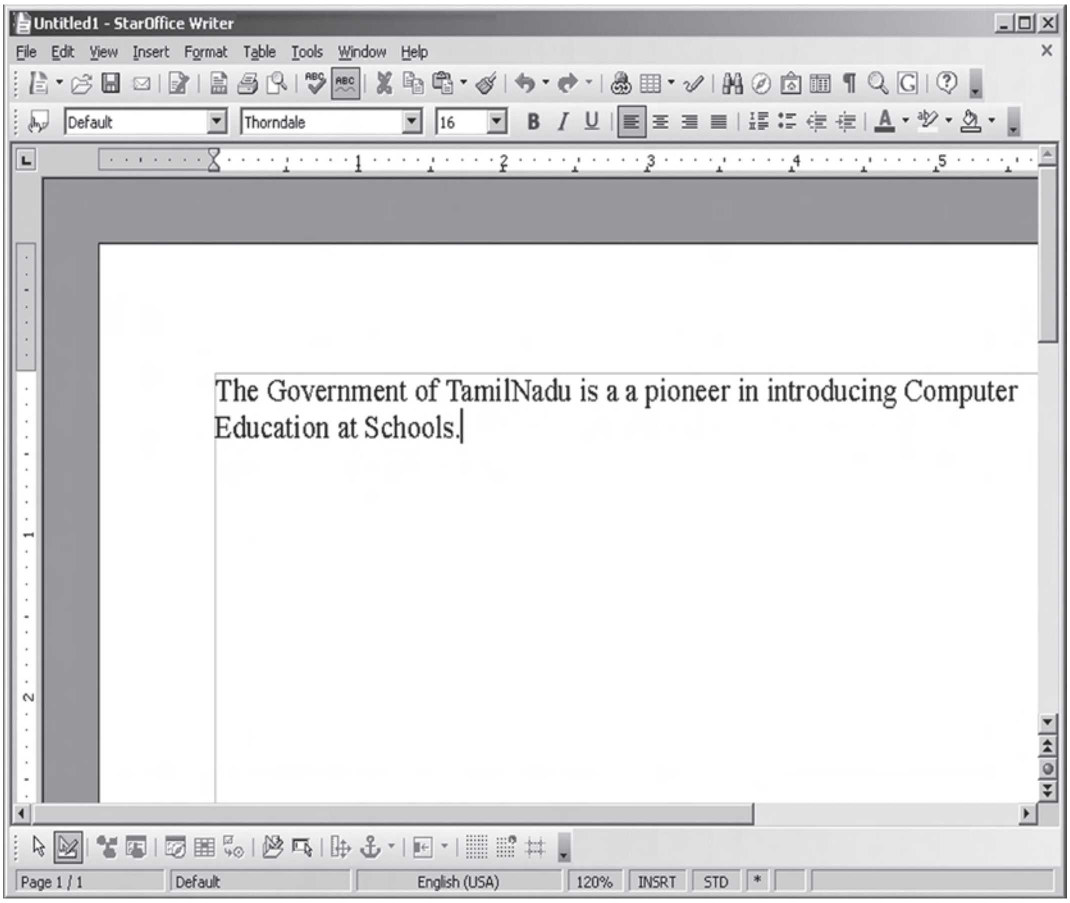
1.9.1 சுட்டெலியின் மூலம் உரையைத் தேர்ந்தெடுத்தல்

1. செருகும் இடத்தை தேர்ந்தெடுக்கப் படவேண்டிய உரையின் தொடக்கத்தில் வைக்க வேண்டும்.
2. சுட்டெலியின் வலது பொத்தானை அழுத்திப் பிடித்தவாறு உரையின் மீது நகர்த்த வேண்டும்.
3. உரை தேர்ந்தெடுக்கப்பட்ட பின் பொத்தானை விட்டு விட வேண்டும். இப்பொழுது உரை படம் 1.5–இல் காட்டியவாறு தோன்றும்.
4. தவறாகத் தேர்ந்தெடுக்கப்பட்ட உரையைக் கைவிடுவதற்கு தேர்ந்தெடுக்கப்பட்ட உரைக்கு வெளியே ஒரு முறை கிளிக் செய்ய வேண்டும்.

1.9.2 விசைப் பலகையின் மூலம் உரையைத் தேர்ந்தெடுத்தல்

விசைப் பலகையின் உதவியுடன் உரையைத் தேர்வு செய்ய கீழ்க்காணுமாறு செய்க:

1. செருகும் சுட்டியை தேர்ந்தெடுக்கப்பட வேண்டிய உரையின் தொடக்கத்தில் வைக்க வேண்டும்.
2. Shift பொத்தானை அழுத்தியவாறு நகர்வுப் பொத்தான்களை பயன்படுத்தி தேவையான உரையை உயர்த்திக் காட்ட வேண்டும்.
3. தேவையான உரை தேர்வு செய்யப்பட்டபின் shift பொத்தானை அழுத்துவதை விட்டு விடவும்.



படம் 1.5 உரையைத் தேர்ந்தெடுத்தல்

1.9.3. உரையைத் தேர்ந்தெடுப்பதற்கான குறுக்கு வழிகள்

செய்ய வேண்டிய செயல்	எதைத்தேர்ந்தெடுக்க
சொல்லின் மீது இரண்டு முறை கிளிக் செய்தல்.	ஒரு வார்த்தையை தேர்ந்தெடுக்க
வரிக்குத் தொடக்கத்தில் சுட்டியை எடுத்துச் சென்று, Shift + End பொத்தான்களை ஒரு முறை கிளிக் செய்தல்	ஒரு குறிப்பிட்ட வரியைத் தேர்ந்தெடுக்க
Ctrl+ A பொத்தான்களை அழுத்துதல்	முழு ஆவணத்தை தேர்ந்தெடுக்க

1.10 உரையை நகர்த்துதல்

தேர்ந்தெடுக்கப்பட்ட உரையை மிகவும் எளிதாக வேண்டிய இடத்தில் வெட்டவும் ஒட்டவும் செய்யலாம். இதற்கு கீழே குறிப்பிட்டுள்ளவாறு செய்க :

1. முதலில் நகர்த்தப்பட வேண்டிய உரையை தேர்ந்தெடுக்க வேண்டும்.
2. பிறகு, Edit → cut கட்டளையை தேர்ந்தெடுக்க வேண்டும். அல்லது கருவிப் பட்டையில் பணிக்குறியைத் தேர்ந்தெடுக்க வேண்டும்.

3. அதன் பிறகு, செருகும் இடத்தை உரையை எங்கு ஒட்ட வேண்டுமோ அங்கு வைக்க வேண்டும்.

4. இறுதியாக Edit → Paste கட்டளையை அல்லது கருவிப் பட்டையில் பணிக்குறியை தேர்ந்தெடுத்து புதிய இடத்தில் தேர்ந்தெடுக்கப்பட்ட உரையை ஒட்ட வேண்டும்.

இந்த முறையின் மூலம் ஒரு ஆவணத்தில் இருந்து இன்னொரு ஆவணத்துக்குக் கூட வெட்டி ஒட்டும் பணியைச் செய்ய முடியும்.

கீழ்க்காணும் குறுக்கு வழியைப் பயன்படுத்தியும் உரையை நகர்த்தலாம்.

Ctrl + X	→	வெட்ட
Ctrl + V	→	ஒட்ட

1.11 உரை நகல் எடுத்தல்

1. வேண்டிய உரையைத் தேர்ந்தெடுக்க வேண்டும்.

2. Edit → Copy கட்டளையை தேர்ந்தெடுக்க வேண்டும். அல்லது பணிக்குறியை கிளிக் செய்ய வேண்டும்.

3. செருகும் இடத்தை தேர்ந்தெடுத்து அந்த இடத்தில் பணிக்குறியை கிளிக் செய்ய வேண்டும்.

உரை நகல் எடுப்பதற்கான குறுக்கு வழிகள்

சாவிச் சேர்மானங்கள்		**செயல்**
Ctrl + C	→	நகல் எடுக்க
Ctrl + V	→	ஒட்ட

1.12 வேண்டிய சொல் அல்லது உரைப் பகுதியைக் கண்டு பிடித்து மாற்றுதல்

Star Office Writer பயன்பாட்டில் உள்ள கண்டுபிடித்து மாற்றுவதற்கான வசதியைப் பயன்படுத்தி, ஒரு உரையில் உள்ள ஒரு சொல் அல்லது உரைப்பகுதியை கண்டு பிடித்து, அந்த சொல் அல்லது உரைப்பகுதியை உரைத்தொகுப்பில் வரும் எல்லா இடங்களிலும் மாற்றி அமைக்கலாம்.

வேண்டிய சொல் அல்லது உரையைத் தேடி மாற்றுவதற்கான வழிகள் பின்வருமாறு :

1. Edit → Find & Replace கட்டளைகளைத் தேர்வு செய்க. உரையாடல் பெட்டி படம் 1.6–இல் காண்பிக்கப்பட்டவாறு தோன்றும்.

2. Search for உரைப் பெட்டியில் தேட வேண்டிய சொல்லை அல்லது உரைப்பகுதியை தட்டச்சு செய்க.

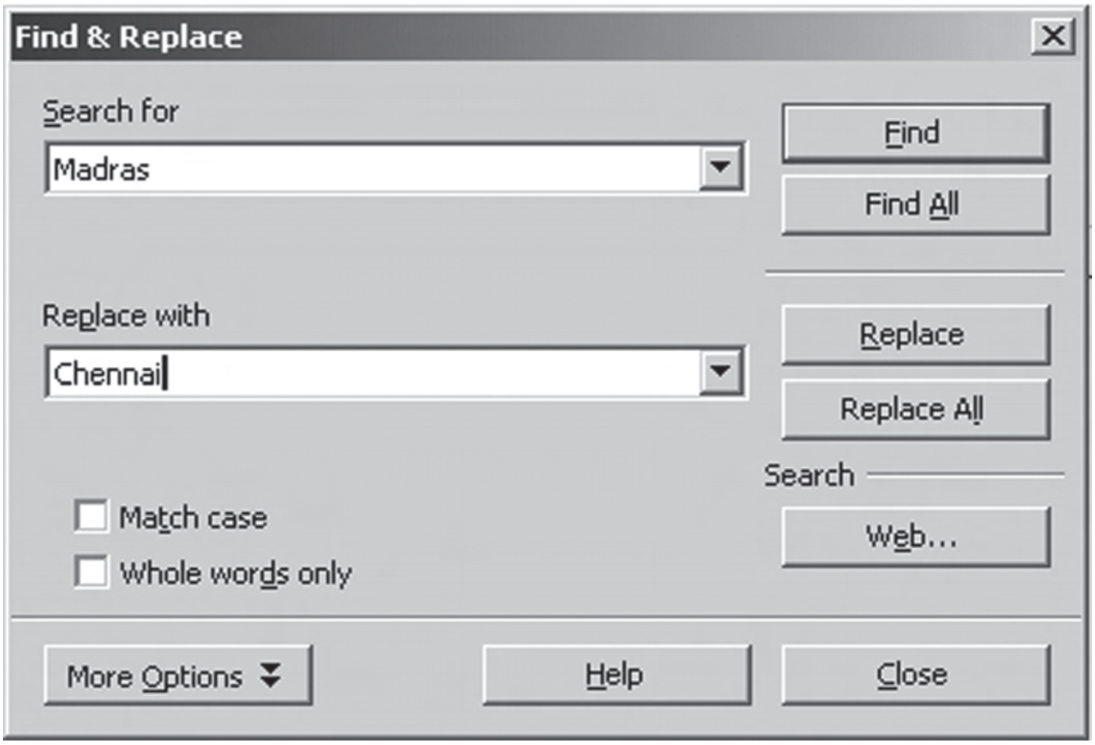
3. Replace with பெட்டியில் மாறி வர வேண்டிய சொல்லை அல்லது உரைப்பகுதியை தட்டச்சு செய்க.

4. தேடுவதற்கு Find பொத்தானை கிளிக் செய்யவும்.

5. Star Office Writer தேட வேண்டிய சொல் அல்லது உரைப்பகுதியை முதல்முறையாகக் கண்டு பிடித்தவுடன் கீழ்க்காணும் ஏதேனும் ஒரு செயலைச் செய்யவும்.

- கண்டுபிடிக்கப்பட்ட அந்த சொல் அல்லது உரைப்பகுதியை மட்டும் Replace with பெட்டியில் தட்டச்சு செய்த சொல் அல்லது உரைப்பகுதியைக் கொண்டு மாற்ற Replace பொத்தானை கிளிக் செய்யவும்.
- கண்டுபிடிக்கப்பட்ட சொல் அல்லது உரைப்பகுதியை உரைத்தொகுப்பில் வரும் எல்லா இடங்களிலும் மாற்றி அமைக்க Replace All என்ற கட்டளையை கிளிக் செய்யவும்.
- கண்டுபிடித்த சொல் அல்லது உரைப்பகுதியைத் தவிர்க்க Find கட்டளையை மீண்டும் கிளிக் செய்யவும்.

உரையாடல் பெட்டியை மூடுவதற்கு Close என்ற பொத்தானை கிளிக் செய்ய வேண்டும்.



படம் 1.6 தேடுதல் மற்றும் மாற்றியிடுதல் உரையாடல் பெட்டி

செய்து பார்

1. Exercise 1 என்ற ஸ்டார் ஆஃபிஸ் ரைட்டர் ஆவணத்தின் முதலிலும், இடையிலும், இறுதியிலும் கூடுதல் பகுதிகளைச் சேர்க்கவும்.
2. Delete மற்றும் Backspace பொத்தானைப் பயன்படுத்தி ஆவணத்தைத் திருத்தவும்.
3. Insert விசையைப் பயன்படுத்தி ஒரு சொல்லைப் புகுத்தவும், அப்பொழுது நிலைமைப் பட்டையில் insert முறைமை எவ்வாறு மாறுகிறது என்று பார்க்கவும்.
4. ஆவணத்தின் ஒரு குறிப்பிட்ட பகுதியை குறுக்குவழி (Shortcut) சுட்டி மற்றும் (cursor) சாவிப்பலகை மூலம் தேர்ந்தெடுக்கவும்.

5. பணிக்குறிகள் மற்றும் விசைப்பலகைக் குறுக்குவழிகள் மூலம் நகல் எடுத்தல், வெட்டுதல், ஒட்டுதல் ஆகிய வேலைகளைச் செய்து பார்க்கவும்.

6. Search and Replace உரையாடல் பெட்டியைப் பயன்படுத்தி ஒரு குறிப்பிட்ட சொல்லை மாற்றிப் பார்க்கவும்.

சுருக்கம்

- ஆவணத்தில் புதியதாக ஒரு உரையைச் சேர்த்தல் அல்லது ஏற்கனவே உள்ள உரையில் மாற்றங்களைச் செய்தல் ஆகிய வேலைகளுக்கு உரை பதிப்பித்தல் (text editing) என்று பெயர்.
- ஆவணத்தின் பல பகுதிகளுக்கும் செல்ல சுட்டெலியையோ அல்லது விசைப்பலகை குறுக்கு வழிகளையோ பயன்படுத்தலாம்.
- வேண்டிய உரைப்பகுதியைத் தேர்வு செய்வதற்கும் சுட்டெலி அல்லது விசைப்பலகை குறுக்கு வழியைப் பயன்படுத்தலாம்.
- நகல் எடுத்தல் (copy), வெட்டுதல் (cut), ஒட்டுதல் (paste), தேடுதல் மற்றும் மாற்றியிடுதல் ஆகியவை முக்கியமான பதிப்பித்தல் பணிகளாகும்.

பயிற்சி

I. **கோடிட்ட இடத்தை நிரப்புக.**

1. உரைப் பகுதியிலுள்ள தடித்த நெடுங்கிடைக் கோட்டுக்கு ________________ என்று பெயர்.

2. ____________ சாவி செருகும் இடத்துக்கு வலது பக்கத்திலுள்ள எழுத்துக்களை அழிக்க உதவுகிறது.

3. ____________ , ____________ தேர்வு மூலம் தேர்வு செய்யப்பட்ட உரையை வெட்டலாம்.

4. ____________ , மற்றும் ____________ என்பவை தேடுதல் மற்றும் மாற்றியிடுதல் உரையாடல் பெட்டியில் உள்ள இரண்டு சேர்மானப் பெட்டிகள் ஆகும்.

5. ____________ பொத்தான் சேர்மானத்தின் மூலம் ஆவணத்தின் இறுதிக்குச் செல்ல முடியும்.

II. **சரியா, தவறா என்று கூறு**

1. உரை தட்டச்சு செய்யும் பொழுது Enter பொத்தானை ஒவ்வொரு வரியின் இறுதியிலும் அழுத்த வேண்டும்.

2. முழு ஆவணத்தையும் தேர்வு செய்வதற்கு Ctrl + A என்ற பொத்தான் சேர்மானத்தைப் பயன்படுத்த வேண்டும்.

3. தேடுதல் மற்றும் மாற்றியிடுதல் உரையாடல் பெட்டி மூலம் ஒரு சொல்லைக் கண்டுபிடிக்க முழுச் சொல்லையும் தட்டச்சு செய்ய வேண்டியதில்லை.

4. தேர்வு செய்யப்பட்ட ஒரு சொல்லை வேண்டிய இடத்தில் ஒட்டுவதற்கு insert பொத்தான் பயன்படுகிறது.

5. Ctrl + X விசைப்பலகை குறுக்குவழி மூலம் தேர்வு செய்யப்பட்ட உரையை நகலெடுக்க முடியும்.

III. **கீழ்க்கண்ட வினாக்களுக்கு விடையளி**

1. உரை பதிப்பித்தல் என்றால் என்ன ?

2. ஒரு ஆவணத்தில் தேவைப்படும் உரைப்பகுதியை எவ்வாறு தேர்வு செய்யலாம் ?

3. insert முறைமையிலிருந்து Type Over முறைமைக்கு எவ்வாறு மாறலாம் ?

4. ஆவணத்திலுள்ள உரையில் ஒரு குறிப்பிட்ட சொல்லைத் தேடும் வழிகளைக் கூறு.

5. உரையிலுள்ள ஒரு சொல்லுக்குப் பதிலாக பிறிதொரு சொல்லைப் புகுத்தும் வழிமுறைகள் யாவை ?

அதிகாரம் 2

உரை வடிவூட்டல்

வடிவூட்டம் (format) செய்யப்படாத ஆவணத்தைப் படிக்கும்போது, அது படிப்பவருக்கு எளிமையாக இராது. ஆனால் முக்கியமான பகுதிகள் தடிமனான எழுத்துக்களிலும், சில குறிப்பிட்ட சொற்கள் உயர்த்திக் காட்டப்பட்டும், மேற்கோள்கள் மற்றும் விளக்கங்கள் மாறுதலான எழுத்துக்களிலும் இருந்தால் அது படிப்பவரின் ஆர்வத்தைத் தூண்டுவதாக அமையும். ஒரு ஆவணத்திலுள்ள உரைக்கு பல விதமான வடிவூட்டங்களைக் கொடுக்கமுடியும்.

2.1 வடிவூட்டல் விருப்பங்கள் (formatting options)

ஸ்டார் ஆஃபிஸ் ரைட்டரின் மூலம் உரையை தடித்த எழுத்துக்களாகவோ (bold), சாய்ந்த எழுத்துக்களாகவோ (italics) அல்லது எழுத்துக்களின் அளவு (size) மற்றும் வகைகளையும் (fonts) மாற்றி அமைத்துக் கொள்ள முடியும்.

வடிவூட்டம் (format) என்ற பட்டிப் பட்டைத் தேர்வு மூலம் ஏறக்குறைய எல்லா வடிவூட்டத் தேர்வுகளையும் பெற முடியும். பொதுவாகப் பயன்படும் தேர்வுகளுக்கென்று தனியாகப் பொத்தான்கள் உள்ளன. ஆனால் இந்த பொத்தான்களைப் பயன்படுத்துவதற்கு முன்னால் அவற்றை எந்த உரைப் பகுதியின் மீது பயன்படுத்துகிறோமோ அதைத் தேர்வு செய்ய வேண்டும். தேவையான உரையைத் தேர்வு செய்தபின் தேவைக்கேற்ப கீழ்க்கண்ட பொத்தான்களில் ஒன்றை கிளிக் செய்ய வேண்டும்.

உரையை தடிப்பாக்குவதற்கு **B** (bold) பொத்தானை கிளிக் செய்ய வேண்டும்.

உரையை சாய்ந்த எழுத்துக்களில் மாற்றுவதற்கு *I* பொத்தானைக் கிளிக் செய்ய வேண்டும்.

உரையை அடிக்கோடிடுவதற்கு U என்ற பொத்தானைக் கிளிக் செய்ய வேண்டும்.

Format → character என்ற கட்டளையைத் தேர்வு செய்து கிடைக்கும் Style பட்டிப் பெட்டியில் ஒரு தேர்வைச் செய்தும் மேற்கூறிய பணிகளைச் செய்ய முடியும்.

ஒரு குறிப்பிட்ட சொல் அல்லது உரைப்பகுதி ஏற்கனவே தடித்த, சாய்ந்த அல்லது அடிக்கோடிடப்பட்ட எழுத்துக்களில் இருந்தால் அந்த உரைப் பண்பை எளிதாக நீக்க முடியும். அந்தப் பகுதியை மீண்டும் தேர்வு செய்து பின்னர் தேவையான பொத்தானைக் கிளிக் செய்ய வேண்டும். உதாரணமாக, தடித்த எழுத்துக்களில் உள்ள ஒரு வார்த்தையை சாதாரண எழுத்துக்களில் மாற்றுவதற்கு அந்த வார்த்தையைத் தேர்வு செய்து **B** என்ற பொத்தானை கிளிக் செய்ய வேண்டும்.

இதற்கு மாறாக Ctrl + B, Ctrl + I, Ctrl + U சாவிச் சேர்மானங்களின் மூலம் உரையை முறையே தடித்த, சாய்ந்த அல்லது அடிக்கோடிட்ட எழுத்துப் பண்புக்கு மாற்றலாம்.

2.1.1 எழுத்து வகையை மாற்றுதல் (Fonts)

ஒவ்வொரு எழுத்து வகையிலும் எழுத்துக்களும், எண்களும் பிறிதொரு எழுத்து வகையிலிருந்து உருவ அமைப்பில் மாறுபட்டிருக்கும். Times New Roman போன்ற எழுத்து வகைகள் பொதுவாக எல்லா அலுவலக ஆவணங்களிலும் பயன்படுத்துகிறனர். ஆனால் FAJITA போன்ற எழுத்து வகைகள் அலங்கார எழுத்துக்களைக் கொண்டிருக்கும். இவற்றைச் சொந்தப் பணிக்கும், அழைப்பிதழ் முதலியன அச்சிடுவதற்கும் பயன்படுத்தலாம். ஆனால்

Symbol, Windings போன்ற எழுத்து வகைகள் எழுத்துக்களாக இல்லாமல் குறியீடுகளாக இருக்கும். அந்த எழுத்து வகைகளின் மூலம் சில சிறப்பு எழுத்துக்களை ஆவணத்தில் நுழைக்க முடியும்.

சரியான எழுத்து வகையைத் தேர்ந்தெடுத்து, சொல் தொகுக்கும் பணியில் ஒரு திறமையான செயலாகும். வேண்டிய உரையைத் தேர்வு செய்து கணினியில் இருக்கும் எந்தவொரு எழுத்து வகையையும் அதற்குப் பயன்படுத்தலாம். எழுத்து வகை மென்பொருட்களை (font packages) விலைக்கு வாங்கியும் விருப்பமான எழுத்து வகைகளைச் சேர்த்துக் கொள்ளலாம்.

எழுத்து வகையை மாற்றும் முறை

Times New Roman என்பதற்குப் பக்கத்தில் உள்ள கீழ் அம்புக்குறியைக் கிளிக் செய்து பின்னர் கிடைக்கும் எழுத்து வகைகளில் ஒன்றைத் தேர்வு செய்ய வேண்டும்.

2.1.2 எழுத்து வகையின் அளவு

எழுத்தின் வகையைப் போல அதன் அளவும் மிக முக்கியமானதாகும். ஒரு சட்ட ஆவணத்துக்கு, விளம்பர அறிவிப்புக்கும் பயன்படுத்தப்படும் எழுத்தின் அளவு வேறானதாக இருக்கும். அதேபோல முக்கிய உரைப் பகுதியில் உள்ள எழுத்தின் அளவும், அந்த உரையின் கீழ்ப் பகுதியிலுள்ள அடிக்குறிப்பும் ஒரே அளவானதாக இருக்க முடியாது. எழுத்து வகையின் அளவை புள்ளிகளில் அளக்கிறோம். ஒரு அங்குலத்துக்கு 72 புள்ளிகள் கொண்ட ஒரு எழுத்தின் அளவு (72 x 1/72) = 1/6 அங்குலமாகும்.

எழுத்தின் அளவை மாற்றும் முறை

12 என்பதின் அருகில் உள்ள கீழ் நோக்கிய அம்புக்குறியை கிளிக் செய்து தேவையான புள்ளி அளவைத் தேர்வு செய்ய வேண்டும்.

2.1.3 எழுத்து வகையின் நிறத்தை மாற்றுதல்

தேர்ந்தெடுக்கப்பட்ட உரைக்கு வேறுவேறு வண்ணங்களைச் சேர்க்க முடியும். வண்ண அச்சுப் பொறிகள் பெரும் வரவேற்பைப் பெற்று வருகின்றன. வண்ண அச்சுப் பொறிகளின் துணைகொண்டு ஆவணங்களைப் பல வண்ணங்களில் அச்சிட முடியும். வண்ண அச்சுப்பொறி இல்லாவிட்டால் ஆவணம் கறுப்பு வெள்ளையில்தான் அச்சிடப்படும்.

நிறத்தை மாற்றும் முறை

உரைக்கு வேறு வண்ணத்தைச் சேர்ப்பதற்கு என்ற பணிக்குறியை கிளிக் செய்து பின்னர் வேண்டிய வண்ணத்தைச் சேர்க்கலாம். இந்தக் பணிக்குறியின் மீது கிளிக் செய்து பொத்தானை அழுத்திப் பிடித்தவாறு இருந்தால் ஒரு வண்ணத்தட்டு திரையிடப்படும். அதில் வேண்டிய வண்ணத்தை கிளிக் செய்தால் தெரிவு செய்யப்பட்ட உரை அந்த வண்ணத்துக்கு மாற்றப்படும்.

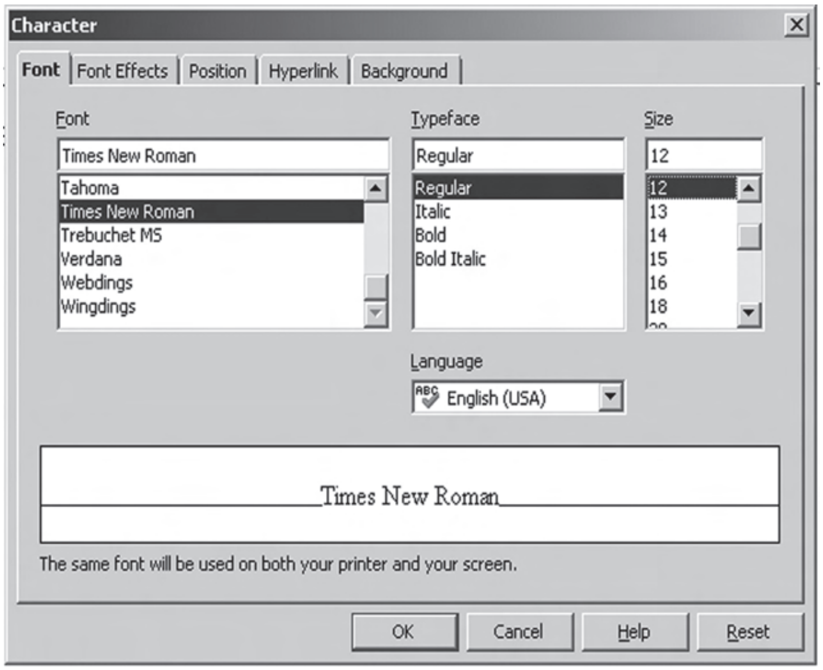
2.1.4 உரையை உயர்த்திக் காட்டுதல் (Highlighting)

முக்கியமான உரைப்பகுதியை படிப்பவரின் கவனத்துக்குக் கொண்டு வருவதற்கு அதனை உயர்த்திக் காட்டுவது மிகவும் பயனுள்ளதாகும். பொதுவாக ஒரு புத்தகத்தில் முக்கியமான பகுதியைப் படிக்கும்பொழுது படிப்பவர் ஒரு மஞ்சள் வண்ண எழுதுகோலின் துணைகொண்டு அதை அடிக்கோடிடுவது வழக்கம். அதைப் போலவே ஸ்டார் ஆஃபிஸ் ரைட்டரில் உரையை பல வண்ணங்களில் உயர்த்திக் காட்ட முடியும்.

உரையை உயர்த்திக் காட்டும் முறை

தேர்வு செய்யப்பட்ட உரையை உயர்த்திக் காட்ட A என்ற பணிக்குறியைக் கிளிக் செய்து பின் தேவையான வண்ணத்தைக் கிளிக் செய்ய வேண்டும். சுட்டெலியின் பொத்தானைக் கிளிக் செய்து அழுத்திப் பிடித்தவாறு இருந்தால் ஒரு வண்ணத் தட்டு திரையிடப்படும். அதிலிருந்து தேவையான வண்ணத்தைத் தேர்வு செய்து கொள்ளலாம். உயர்த்திக் காட்டப்பட்டுள்ள உரையை சாதாரண உரையாக மாற்றுவதற்கு அதைத் தேர்வு செய்து பின் வண்ணத் தட்டிலிருந்து No fill என்பதைத் தேர்வு செய்ய வேண்டும்.

மேற்சொன்ன அனைத்து வடிவூட்டுதலையும் ஒன்றாகச் செய்வதற்கு Format என்ற பட்டித் தேர்வில் Character என்ற கட்டளையைக் கிளிக் செய்தால் படம் 2.1 இல் காட்டப்பட்டுள்ள உரையாடல் பெட்டி தோன்றும் இந்த உரையாடல் பெட்டி மூலம் பல மாறுதல்களை ஒன்றாகச் செய்வது மட்டுமல்லாமல் அந்த மாறுதல்களின் முன்னோட்டத்தையும் காண முடியும்.



படம் 2.1 உரையாடல் பெட்டி

செய்து பார்

Exercise 1 என்ற ஆவணத்தில் ஒரு குறிப்பிட்ட உரையைத் தேர்வு செய்து கீழ்க்கண்ட பணிகளைச் செய்யவும். தேர்வு செய்யப்பட்ட உரையை தடிமனான எழுத்தாகவும், சாய்வான எழுத்தாகவும், பின்னர் அடிக்கோடிட்ட எழுத்தாகவும் மாற்றவும். அதன் பின்னர் சாதாரண உரையாக மாற்றவும், எழுத்துவகை, அதன் அளவு மற்றும் வண்ணத்தை மாற்றிப் பார்க்கவும்.

2.2 பத்தி இசைவு (Paragraph Alignment)

தனித்தனி சொற்களை வடிவூட்டுதல் செய்தலோடு பத்திகளையும் வடிவூட்டுதல் செய்யலாம். இதில் முக்கியமானது பத்தியின் இசைவை மாற்றுதலாகும். சில நேரம் ஒற்றை வரிகூட ஒரு பத்தியாக இருக்கும். ஒரு பத்திக்குத் தனியாகவோ அல்லது பல பத்திகளுக்கும் ஒன்றாகவே வடிவூட்டலைச் செய்யலாம்.

ஒவ்வொரு முறை நுழைவுப்பொத்தானை அழுத்தும் பொழுதும் அந்தப் பத்திக்கான விருப்பங்கள் (options) அடுத்த பத்திக்கும் அனுப்பப்படுகின்றன. ஒரு பத்தியின் அடையாளத்தை (paragraph marker) நீக்கி விட்டால் அதற்கு பின் உள்ள பத்தியின் வடிவூட்டத்தை அது எடுத்துக் கொள்ளும். ஒரு வரியைத் தட்டச்சு செய்த பின் அதை மையப்படுத்தி நுழைவுப் பொத்தானை அழுத்தினால் அதை அடுத்துள்ள வரியும் மையப்படுத்தப்படும்.

இரட்டை வரி இடைவெளி அமைப்பில் தட்டச்சு செய்த ஒரு பத்தியை அடுத்து ஒரு ஒற்றை வரி இடைவெளி அமைப்பில் தட்டச்சு செய்த பத்தி இருந்த, முதல் பத்தியின் பத்தி அடையாளத்தை நீக்கிவிட்டால் இரண்டாவது பத்தியும், இரட்டைவரி இடைவெளி அமைப்பு உள்ளதாக மாறிவிடும்.

ஸ்டார் ஆஃபிஸ் ரைட்டர் (Star Office writer) மென்பொருளில் தட்டச்சு செய்யும் பொழுது எல்லா உரைகளும் இடது பக்க இசைவிலும் (Left Alignment) வலது பக்கம் இசைவில்லாமலும் (Right Alignment) இருக்கும். சில பத்திகளுக்கு இசைவு மாறுதல் செய்ய வேண்டி வரும். எடுத்துக்காட்டாக அவை வலது பக்க இசைவாகவோ மையப்படுத்தப்பட்டோ அல்லது நேர்த்தி செய்யப்பட்டோ (Justified) இருக்கும். ஆவணத்தை கருவிப் பட்டை மூலம் இசைவு செய்வதற்கு கீழ்க்கண்ட வழிகள் உதவுகின்றன.

1. இசைவு மாற்றம் செய்யப்பட வேண்டிய பத்தியின் உட்புறத்தில் ஒரு முறை கிளிக் செய்ய வேண்டும். பல பத்திகளுக்கு மாற்றம் செய்ய வேண்டியிருந்தால் அவை அனைத்தையும் தேர்ந்தெடுக்க வேண்டும்.

2. பின்னர்,

பணிக்குறியை இடது இசைவுக்கு கிளிக் செய்ய வேண்டும்.

பணிக்குறியை வலது இசைவுக்கு கிளிக் செய்ய வேண்டும்.

பணிக்குறியை மைய இசைவுக்கு கிளிக் செய்ய வேண்டும்.

பணிக்குறியை நேர்த்தி இசைவுக்கு கிளிக் செய்ய வேண்டும்.

To undo the change, click the Undo button or select the Edit → Undo command.
If a different alignment is required later, select the paragraph (s) and then click
another alignment button.

To undo the change, click the Undo button or select the Edit → Undo command.
If a different alignment is required later, select the paragraph (s) and then click another
alignment button.

To undo the change, click the Undo button or select the Edit → Undo command.
If a different alignment is required later, select the paragraph (s) and then click
another alignment button.

To undo the change, click the Undo button or select the Edit → Undo command.
If a different alignment is required later, select the paragraph (s) and then
click another alignment button.

படம் 2.2 பத்தி இசைவு

படம் **2.2** இல் ஒவ்வொரு இசைவுக்கும் எடுத்துக்காட்டு கொடுக்கப்பட்டு உள்ளது.

கீழ்க்கண்ட விசைப்பலகை குறுக்கு வழிகளையும் பயன்படுத்தலாம் (Key board shortcut)

மைய இசைவு	Ctrl + E
இடது இசைவு	Ctrl + L
வலது இசைவு	Ctrl + R
நேர்த்தி இசைவு	Ctrl + J

செய்துபார்

Exercise 1 என்ற ஆவணத்தைத் திறக்கவும். கீழ்க்கண்ட ஆணைகளைச் செயல்படுத்தி ஏற்படும் மாறுதல்களைக் கவனிக்கவும்.

1. பொருத்தமான ஒரு தலைப்பைக் கொடுத்து அதனை மையப்படுத்தவும்.
2. இரண்டாவது பத்தியைத் தேர்வு செய்து அதனை நேர்த்தி செய்யவும்.
3. மூன்றாவது பத்தியைத் தேர்வு செய்து அதனை வலது இசைவு செய்யவும்.
4. உன்னுடைய பெயர், வகுப்பு மற்றும் பள்ளியின் பெயரை பக்கத்தின் அடிப்பகுதியில் தட்டச்சு செய்து பின் அதனை இடது இசைவு செய்யவும்.

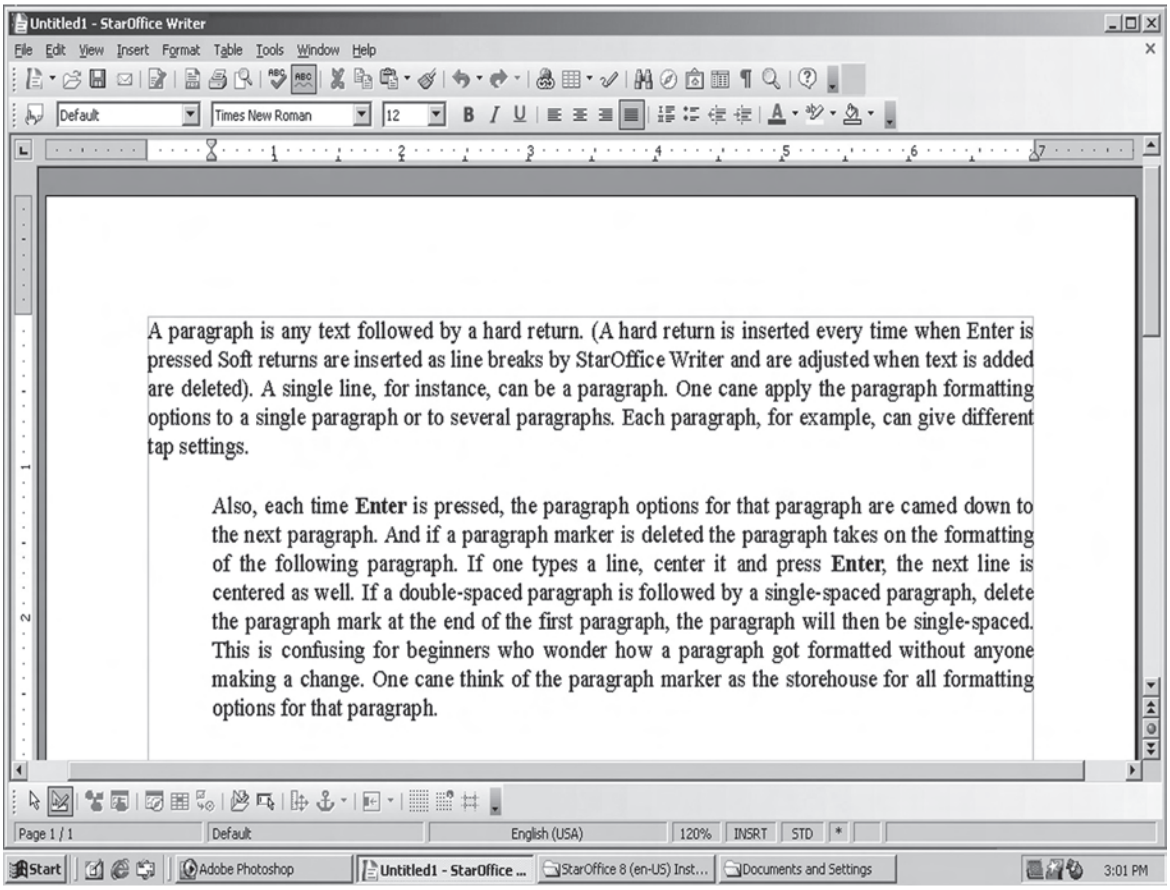
2.3 உரையை உள்தள்ளல் (Indentation)

தலைப்புகளையும் பத்திகளையும் பொறுத்தவரை இசைவு மாற்றங்கள் போதுமானதாக இருக்கும். ஆனால் ஒரு நீண்ட ஆவணத்தைப் படிக்கும் போது படிப்பவருக்கும் ஆர்வமூட்டுவதாகவும், எளிமையானதாகவும் இருக்க வேண்டுமென்றால் இந்தப் பகுதியில் விவரிக்கப்பட்டுள்ள உள்தள்ளல் (Indentation) மிகவும் உதவியானதாக இருக்கும் எடுத்துக்காட்டாக முக்கிய உரைப்பகுதியிலிருந்து ஒரு மேற்கோள் பத்தியை வேறுபடுத்திக் காட்ட வேண்டுமென்றால், அதனைத் தனியே இடது ஓரத்திலிருந்து உள்தள்ளிக் காண்பிக்கலாம். இன்னொரு தேர்வாக, தொங்கும் உள்தள்ளல் (hanging Indent) என்பதனை எண் வரிசைக்குப் (Numbered List) பயன்படுத்தலாம்.

உள்தள்ளலின் அளவைத் தேவைக்கேற்றவாறு மாற்றிக் கொள்ளலாம். மேலும் அதற்காக வடிவூட்டல் கருவிப்பட்டையையோ அல்லது பத்தி உரையாடல் பெட்டியையோ பயன்படுத்தலாம்.

2.3.1 உரையைக் கருவிப்பட்டையின் மூலம் உள்தள்ளல்

ஒரு பத்தியை மற்ற உரைப் பகுதியிலிருந்து சற்றே இடப்புறமாக உள்தள்ளுவதற்கு (left indentent) கருவிப்பட்டையைப் பயன்படுத்தலாம். என்ற பணிக்குறியை கிளிக் செய்தால் உள்தள்ளலை அதிகப்படுத்தலாம். ஒரு முறை கிளிக் செய்தால் இடது ஓரத்திலிருந்து ½ அங்குலம் உள்ளே தள்ளப்படும். அந்தப் பொத்தானை மேலும் ஒரு முறை கிளிக் செய்தால் மேலும் ½ அங்குலம் உள்ளே தள்ளப்படும். படம் **2.3** இல் அவ்வாறு உள்ளே தள்ளப்பட்ட ஒரு பத்தி காண்பிக்கப்பட்டுள்ளது.



படம் 2.3 உள்தள்ளலை அதிகப்படுத்தும்

உள் தள்ளப்பட்ட உரையை மீண்டும் பழைய நிலைக்குக் கொண்டு வருவதற்கு உள்தள்ளலைக் குறைத்தல் பொத்தானைக் கிளிக் செய்ய வேண்டும்

2.3.2 பத்தி உரையாடல் பெட்டி மூலம் உரையை உள்தள்ளுதல்

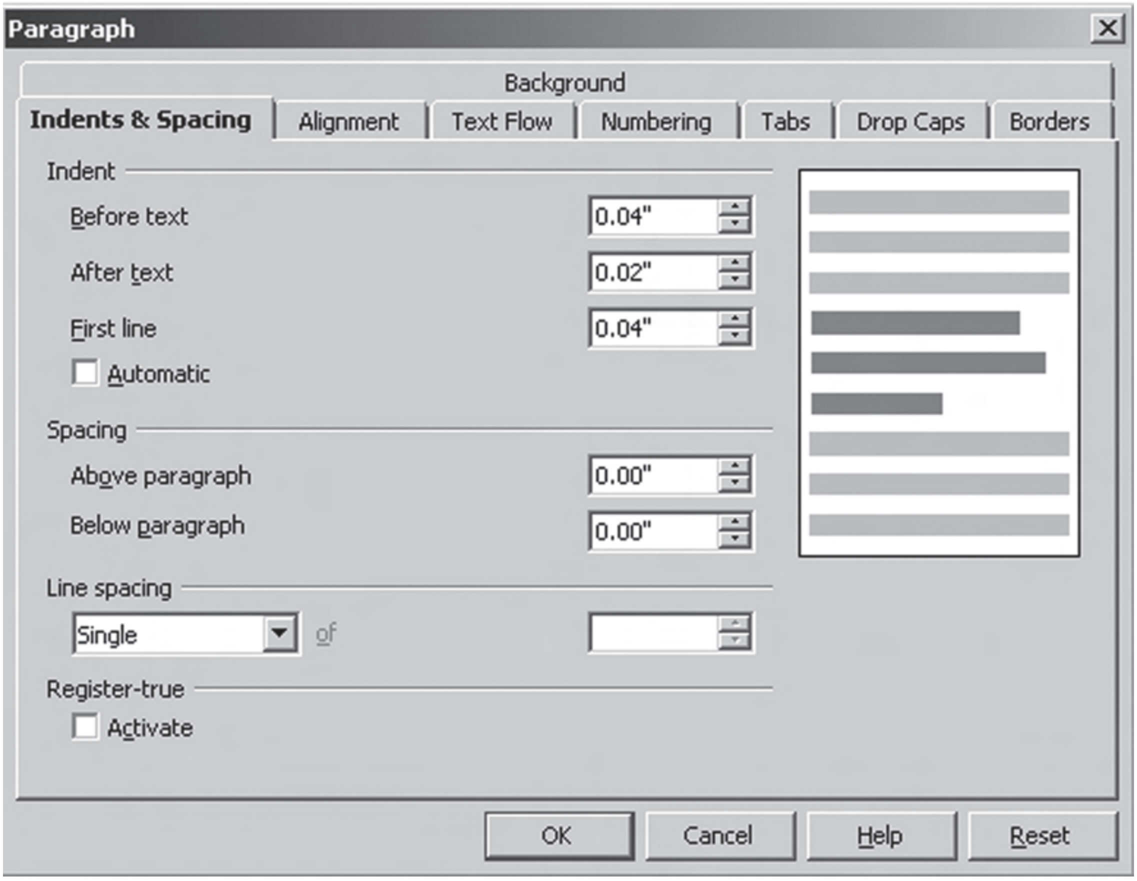
உள்தள்ளலைக் அதிகமாக்குதல் பொத்தான்கள் (Increase Indent button) மற்றும் உள்தள்ளலைக் குறைத்தல் பொத்தான்கள் (Decrease Indent button) உரையை இடப்புற ஓரத்திலிருந்து உள்தள்ளுவதற்கு உதவியாக இருக்கும், உரையை இடது மற்றும் வலது புறத்திலிருந்து உள்தள்ளல் செய்வதற்கும், அல்லது சிறப்பு வகையான உள்தள்ளலுக்கும் வேறு ஒரு வழி உள்ளது. இதற்காக பத்தி உரையாடல் பெட்டி உதவியாக இருக்கும்.

அதற்கான வழி முறைகள் பின்வருமாறு

1. எந்தப் பத்தியை உள்தள்ள வேண்டுமோ அதனுடைய ஆரம்பத்துக்கு செருகும் இடத்தை நகர்த்த வேண்டும்.

2. Format → Paragraph என்ற கட்டளையைத் தேர்வு செய்ய வேண்டும்.

3. பத்தி உரையாடல் பெட்டியில் Indents and Spacing என்ற தொகுதியை படம் 2.4 இல் காட்டியவாறு கிளிக் செய்ய வேண்டும்.



படம் 2.4 Indent and Spacing உரையாடல் பெட்டி

4. கீழ்க்கண்ட ஏதாவது ஒன்றைச் செய்யவும்

இடது புறமிருந்து உள்தள்ள From left என்ற சுழல் பெட்டியில் உள்தள்ள வேண்டிய அளவைத் தட்டச்சு செய்ய வேண்டும். அல்லது சுழல் அம்புக்குறியைப் பயன்படுத்தி ஒரு மதிப்பைத் தேர்வு செய்யலாம் இதே முறையில் வலது புறமிருந்து உள்தள்ளலையும் செய்யலாம்.

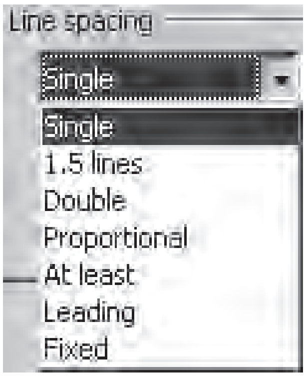
5. OK பொத்தனைக் கிளிக் செய்ய வேண்டும்.

First line என்ற தேர்வைப் பயன்படுத்தி பத்தியின் முதல் வரியை மட்டும் உள்ள தள்ள முடியும். சுழல் அம்புக் குறியைக் கொண்டு ஒரு நேர் மதிப்பைக் (positive value) குறிப்பிட்டால் முதல் வரி உள்தள்ளப்படும். ஒரு எதிர் எண்ணைப் பயன்படுத்தினால் தொங்கும் உள்தள்ளல் உருவாகும். அதாவது முதல் வரியானது பத்தியின் இதர வரிகளைக் காட்டிலும் வெளியே தள்ளிக் கொண்டிருக்கும்.

செய்து பார்

கீழ்க்கண்ட பத்திகளை பட்டிப் பட்டை குறும்படங்கள் மற்றும் பட்டித் தேர்வுகள் மூலம் கொடுத்தவாறே தட்டச்சு செய்யவும்.

2.4 வரி இடைவெளியை மாற்றுதல்



படம் 2.5 வரி இடைவெளித் தேர்வுகள்

கொடாநிலையாக ஸ்டார் ஆஃபிஸ் ரைட்டர் ஆவணங்கள் ஒரு வரி இடைவெளியில் இருக்கும். பெரும்பாலான கடிதங்கள் மற்றும் ஆவணங்களுக்கு இது சரியாக இருக்கும். ஆனால் ஒரு சில ஆவணங்களுக்காக வேறு வரி இடைவெளி அமைப்பு வேண்டியதாக இருக்கும். படம் 2.5 ல் காட்டியவாறு 1 வரி, 1.5 வரி, 2 வரி இடைவெளி அமைப்பும் மற்றும் அளவுக்கேற்ற இடைவெளி(proportional), குறைந்தபட்ச இடைவெளி(At least), தலைமை(Leader), மாறா இடைவெளி (Fixed) ஆகிய தேர்வுகளும் உள்ளன. வரி இடைவெளியை கீழ்க்கண்டவாறு மாற்றலாம்.

1. தேவையான பத்தியைத் தேர்வு செய்ய வேண்டும்.
2. Format → Paragraph என்பதைத் தேர்வு செய்ய வேண்டும்.
3. தேவையானால் Indents and spacing தொகுதியைத் தேர்வு செய்ய வேண்டும்.

   படம் 2.4 இல் காட்டிய உரையாடல் பெட்டி தோன்றும்.
4. வரி இடைவெளி கீழிறங்கு பட்டிப் பெட்டியைத் தெரியச் செய்து தேவையான வரி இடைவெளியைத் தேர்வு செய்ய வேண்டும்.
5. OK பொத்தானைக் கிளிக் செய்ய வேண்டும்.

செய்து பார்

Exercise 1 என்ற ஆவணத்தைத் திறக்கவும். வரி இடைவெளியை 1.5 வரி இடைவெளியாகவும், இரட்டை வரி இடைவெளியாகவும் மாற்றவும் பின் நடைபெறும் மாற்றத்தைக் கவனிக்கவும்.

2.5 புல்லட் குறி மற்றும் எண் வரிசையை உருவாக்குதல்

ஆவணத்தில் ஒரு பொருளைப் பற்றிய பத்தி நீண்டதாக இருந்தால் படிப்பவர் அதிலுள்ள முக்கியச் செய்தியை எளிதாகப் பார்க்க முடியாது. இதன் பொருட்டு, முக்கியக் குறிப்புகளையும், செய்திகளையும் வரிசையிடுவதற்கு புல்லட்டுகளும், எண்வரிசையும் பயன்படுத்தப்படுகிறது.

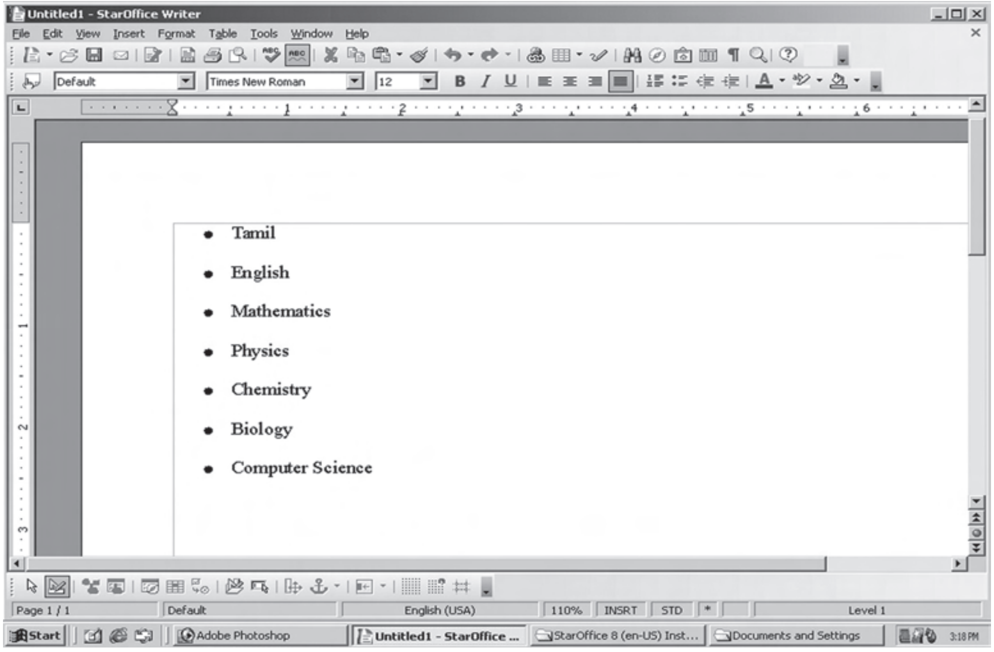
முக்கியத் தலைப்புகளையும், செய்திகளையும் மற்ற உரைப் பகுதியிலிருந்து பிரித்துக் காண்பிப்பதற்கு புல்லட்டுகள் உதவுகின்றன. வரிசையாக உள்ள ஒவ்வொரு உறுப்புக்கும் முன்பு ஒரு புல்லட் குறி இடப்பட்டு அது உள்தள்ளப்படுகிறது. ஏதாவது ஒரு கட்டளையை வரிசையாகத் தர வேண்டுமென்றாலோ அல்லது ஒரு பட்டியல் உறுப்புகளை ஒன்றன்பின் ஒன்றாகக் கொடுப்பதற்கோ, எண் வரிசை (Numbered List) பயன்படுகிறது. ஸ்டார்ஆஃபிஸ் ரைட்டர் தானாகவோ, ஒரு பட்டியலை எண் வரிசையிட்டு உள்தள்ளுகிறது.

2.5.1 புல்லட் குறியிடப்பட்ட வரிசையை உருவாக்குதல்

வடிவூட்டல் கருவிப் பட்டையில் உள்ள புல்லட் பொத்தானைப் பயன்படுத்துவதன் மூலம்தான் விரைவாக புல்லட் குறியிடப்பட்ட வரிசையை உருவாக்க முடியும். இதற்காகப் பயன்படுத்தப்படும் வழிமுறைகள் பின்வருமாறு:

1. புல்லட் குறியிடப்பட வேண்டிய உரையைத் தேர்வு செய்ய வேண்டும். ஸ்டார் ஆஃபிஸ் ரைட்டர் இப்பொழுது இந்தத் தேர்வின் ஒவ்வொரு பத்திக்கும் புல்லட் குறிகளைச் சேர்க்கும். ஆனால் ஒவ்வொரு வரிக்கும்,காலி வரிகளுக்கும் புல்லட் குறி சேர்க்கப்பட மாட்டாது.

2. பொத்தானை கிளிக் செய்ய வேண்டும். படம் 2.6 இல் காட்டியவாறு புல்லட் இடப்பட்ட பட்டியல் தோன்றும்.



படம் 2.6 புல்லட் இடப்பட்ட பட்டி

2.5.2 எண் வரிசைப்பட்டியலை உருவாக்குதல்

ஒரு குறிப்பிட்ட வரிசையில் தோன்றும் பொருட்களுக்கு எண் வரிசைப் பட்டியல் உபயோகமாக உள்ளது. எடுத்துக்காட்டாக ஒரு செயலை செய்து முடிப்பதற்கு கையாள வேண்டிய படிகளை வரிசையாகக் கொடுக்கலாம். ஸ்டார் ஆஃபிஸ் ரைட்டர் எண் வரிசையிடப்பட்ட பட்டியலை தானாக உருவாக்குவதுடன் அந்த பட்டியலை சற்று உள்தள்ளவும் செய்கிறது. இதன் மூலம் அந்தப் பட்டியல் மற்ற உரைகளில் இருந்து வேறுபடுத்திக் காண்பிக்கப்படும். எண் வரிசை பட்டியலில் ஒரு உறுப்பை சேர்க்கவோ நீக்கவோ செய்தால் ஸ்டார் ஆஃபிஸ் ரைட்டர் தானாகவே அதற்கேற்றவாறு எண்களை மாற்றிக் கொள்ளும்.

கீழ்க்கண்ட வழிமுறைகளின் மூலம் எண் வரிசை இடப்பட்ட பட்டியலை உருவாக்கலாம்.

1. எண் வரிசையிடப்பட வேண்டிய உரையைத் தேர்வு செய்ய வேண்டும். ஸ்டார் ஆஃபிஸ் ரைட்டர் இப்பொழுது இந்தத் தேர்வின் ஒவ்வொரு பத்திக்கும் எண் வரிசையைச் சேர்க்கும். ஆனால் ஒவ்வொரு வரிக்கும், காலி வரிகளுக்கும் எண் வரிசை சேர்க்கப்படமாட்டாது.

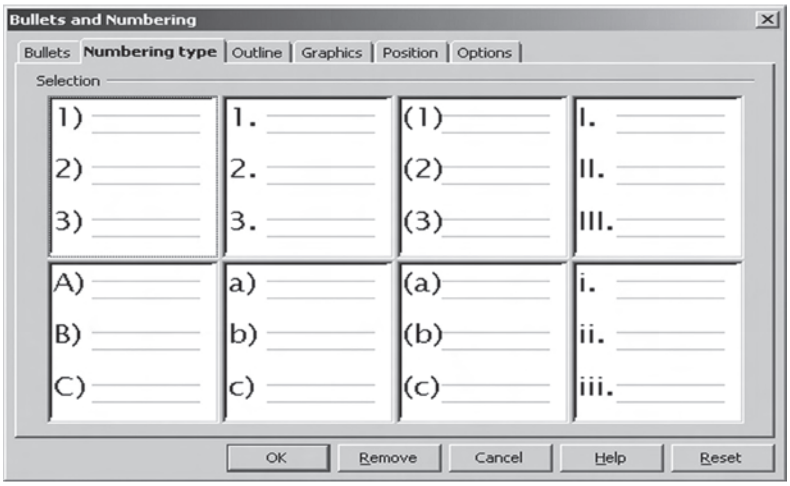
2. பொத்தானை கிளிக் செய்தால் எண் வரிசையிடப்பட்டு (Numbered List) தோன்றும்.

2.5.3 புல்லட் மற்றும் எண் வரிசையை நீக்குதல்

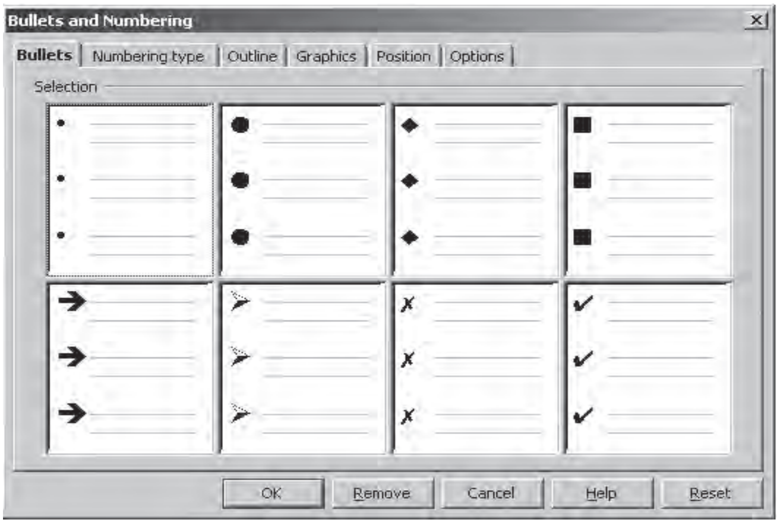
புல்லட் குறியிடப்பட்ட பகுதியைத் தேர்வு செய்து பொத்தானைக் கிளிக் செய்ய வேண்டும். அல்லது எண் குறியிடப்பட்ட பகுதியைத் தேர்வு செய்து பொத்தானைக் கிளிக் செய்ய வேண்டும்.

2.5.4 புல்லட் குறிகளுக்கும் எண் வரிசைக்கும் ஸ்டைல்களைச் சேர்த்தல்

ஸ்டார்ஆஃபிஸ் ரைட்டர் முலம் பலவகையான புல்லட் குறிகளையும் எண் வரிசையையும் பெற முடியும். இதற்காக Format ⟶ Numbering/ bullets என்ற தேர்வைப் பயன்படுத்த வேண்டும். இப்பொழுது படம் 2.7, படம் 2.8 ல் காட்டப்பட்டுள்ள Numbering/ bullets உரையாடல் பெட்டி தோன்றும். இதம் மூலம் நமக்கு தேவையான புல்லட் குறிகளையோ எண் வரிசையோ பெற்றுக் கொள்ளலாம்.



படம் 2.7 (அ) எண் வரிசை உரையாடல் பெட்டி



படம் 2.7 (ஆ) புல்லட் உரையாடல் பெட்டி

செய்து பார்

1. உனக்குப் பிடித்தமான விளையாட்டுக்களின் பட்டியலைத் தயாரிக்கவும், அவற்றை எண் வரிசையிடவும். பின் அதில் மேலும் இரண்டு விளையாட்டுகளைச் சேர்த்து அவற்றிற்கு புல்லட் குறிகள் இடவும்.
2. உன்னுடைய 10 நண்பர்கள் பெயரைத் தட்டச்சு செய்து அவற்றிற்கு புல்லட் குறிகள் இடவும். பின் வேறுவேறான குறிகளை மாற்றிப் பார்க்கவும்.

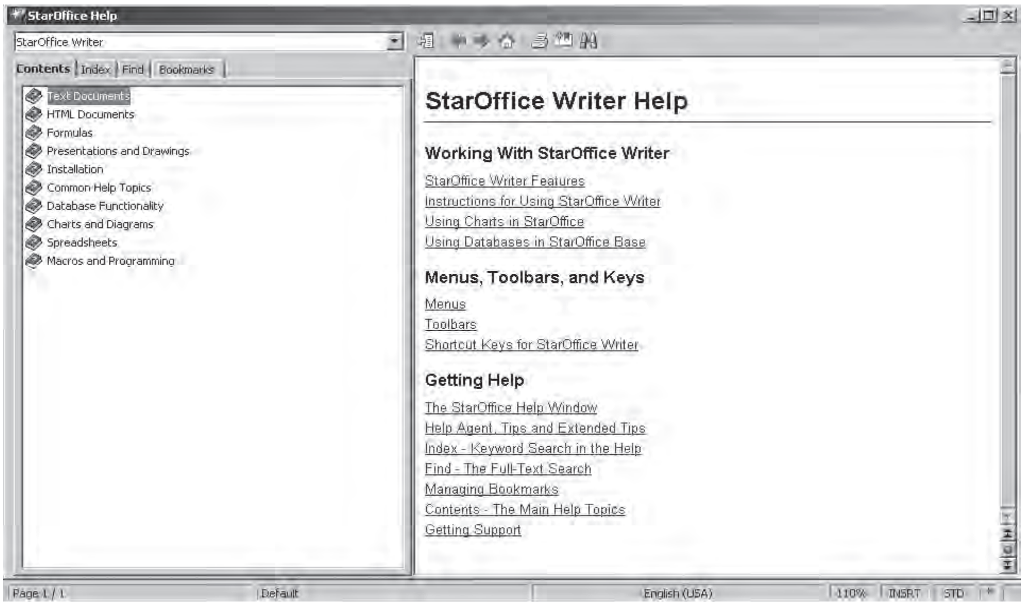
2.6 ஸ்டைல்களைப் (style) பயன்படுத்தி வடிவூட்டுதல்

உரையை வடிவூட்டம் செய்வதற்கு வேறொரு வழியும் உள்ளது. ஒரு உரையை வடிவூட்டம் செய்வதற்கு பல கொடாநிலை (defaults) களையும் பயன்படுத்தலாம். ஒவ்வொரு கொடாநிலைமைக்கும் ஒரு பெயர் இருக்கும். வடிவூட்டலுக்குத் தேவையான இந்த கொடாநிலைமைகளின் தொகுப்புக்கு ஸ்டைல் (style) என்று பெயர். இதற்கு முந்தைய பகுதிகளில் விவரிக்கப்பட்ட வடிவூட்டல் முறைகள் ஒரே முறை செய்ய வேண்டிய வடிவூட்டல்களுக்கு மிகவும் பொருத்தமானதாகும். உதாரணமாக ஒரு சொல்லைத் தடிமனாக்க வேண்டும் என்றால் அதைத் தேர்வு செய்து அதற்கான பணிக்குறியை கிளிக் செய்தால் அது உடனே தடிமனாக்கப்பட்டு விடும்.

ஆனால் ஒவ்வொரு ஆவணமும் பல பக்கங்களைக் கொண்ட பல ஆவணங்களை ஒரு குறிப்பிட்ட விதத்தில் வடிவூட்டம் செய்ய வேண்டுமென்றால் ஸ்டைல் (style) மிகவும் பயனுள்ளதாக இருக்கும் மேலும் ஒரு ஆவணத்துக்குள்ளே பல இடங்களில் ஒரே மாதியான வடிவூட்டலைச் செய்வதற்கும் ஸ்டைல் (style) பயனுள்ளதாக இருக்கும். ஸ்டைலைப் பயன்படுத்துவது என்று முடிவு செய்துவிட்டால் பின்னர் அதற்கென்று தனியாகத் திட்டமிட வேண்டும். ஸ்டைல் என்பதை முதலில் தயாரித்து வைத்துக் கொண்டால் பின்னர் அதனை எளிதாக ஒரு ஆவணத்துக்கோ அல்லது ஒரு குறிப்பிட்ட உரைப்பகுதிக்கோ எளிதாகச் சேர்க்க முடியும். அதிகமான அளவில் வடிவூட்டல் மாறுதல்களைச் செய்யும்பொழுது ஸ்டைல்களின் மேம்பட்ட தன்மை எளிதாக விளங்கும்.

2.7 ஸ்டார் ஆஃபிஸ் உதவி

ஸ்டார் ஆஃபிஸ் தொகுப்பில் ஒரு உடன் நிகழ் (on line) உதவி வசதி உள்ளது. இந்த வசதியை பெறுவதற்கு Help → Content என்ற தேர்வை ஸ்டார் ஆஃபிஸ் desktop துணை கொண்டோ அல்லது ஸ்டார் ஆஃபிஸ் ரைட்டர் பட்டிப் பட்டையில் இருந்தோ பெறலாம். இவ்வாறு செய்தால் படம் 2.8 இல் காட்டிவாறு ஒரு உதவித் திரை (help screen) தோன்றும். இந்த திரையில் Choose help file பெட்டியின் மூலம் தேவையான உதவியைப் பெறலாம். இது உண்மையில் ஒரு பட்டிப்பெட்டியாகும். இதில் பட்டியிலிடப்படும் உறுப்புகள் படம் 2.9 ல் காட்டப்பட்டுள்ளன.

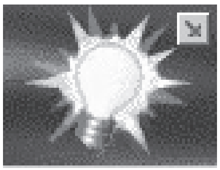


படம் 2.8 உதவித் திரை

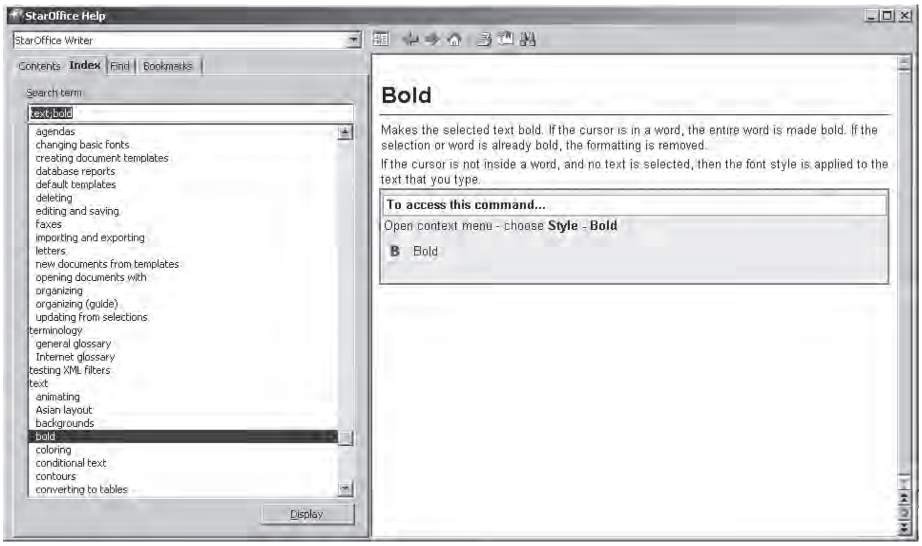
படம் 2.9 பட்டியலிடப்படும் உறுப்புகள்

என்ற தேடல் (Search) பணிக்குறியின் மூலம் தேவையான தலைப்புகளில் உதவியைப் பெறமுடியும்.

ஸ்டார் ஆஃபிஸ் ரைட்டர் ஒரு உதவி முகவர் (help agent) வசதியையும் கொண்டுள்ளது. உதவிப் பட்டியில் கிடைக்கும் இந்த வசதியை உள்ளதாகவோ அல்லது இல்லாததாகவோ (on or off) மாற்றிக் கொள்ளலாம். உதவி முகவர் வசதி உள்ள நிலையில் ஒரு சுழல் உணர் உதவி (context - sensitive help) படம் 2.10 ல் காட்டியவாறு ஒரு சிறிய சன்னல் வழியாகத் திரையில் தோன்றும். உதவி முகவர் சன்னல் திரையின் பொருளடக்கம், முக்கிய சன்னல் திரையில் என்ன வேலை நடைபெறுகிறதோ அதற்கேற்றவாறு மாறிக் கொண்டே இருக்கும். எடுத்துக்காட்டாக ஒரு உரையை தடிமனான எழுத்துக்களில் மாற்றுவதற்கு **B** என்ற பணிக்குறியை கிளிக் செய்யும்பொழுது அந்த பணிக்குறிப் பற்றிய உதவி படம் 2.11 ல் காட்டியவாறு தோன்றும்.



படம் 2.10 சூழல் உணர் உதவி



படம் 2.11 தடிமனான எழுத்துக்களில் மாற்ற பணிக்குறிப் பற்றிய உதவி

சுருக்கம்

- வடிவூட்டலில் வன் வடிவூட்டல் (hard formatting), மென் வடிவூட்டல் (soft formatting) என இரண்டு வகைகள் உள்ளன.
- உரையின் வடிவத்தை தடித்த, சாய்ந்த, அடிக்கோடிடப்பட்ட எழுத்துக்களாக மாற்றுவதும், எழுத்து வகையின் அளவு மற்றும் வண்ணத்தை மாற்றுவதும் பொதுவாகப் பயன்படுத்தப்படும் சில வடிவூட்டல் மாறுதல்களாகும்.
- பத்தி இசைவு கீழ்க்கண்ட நான்கு வகைகளில் செய்யப்படுகிறது.

  1. இடது இசைவு
  2. வலது இசைவு
  3. நேர்த்தி இசைவு
  4. மைய இசைவு

- உள்தள்ளல் வசதி ஆவணங்களை எளிதாக படிக்க உதவி செய்கிறது.
- ஸ்டார் ஆஃபிஸ் ரைட்டர் ஆவணத்தில் வரி இடைவெளிகளை மாற்றுவது மிகவும் எளிதானதாகும்.
- புல்லட் குறி மற்றும் எண் வரிசையிடப்பட்ட பட்டியல்கள் ஆவணத்தின் முக்கிய பகுதிகளைக் காட்ட உதவுகின்றன.
- வரி இடைவெளி, பத்தி இசைவு, உள்தள்ளல் போன்ற வடிவூட்டல் மாறுதல்களை தனித்தனியாக செய்யாமல் ஒரு ஸ்டைலை (Style) உருவாக்கிக்கொண்டால் பெரிய ஆவணங்களுக்கு எளிதாக வடிவூட்டல் செய்யலாம்.
- ஸ்டார் ஆஃபிஸ் ரைட்டரில் உள்ள உடன் நிகழ் உதவி வசதி மூலம் அனைத்து தலைப்புகள் குறித்தும் உதவி பெறமுடியும்.

பயிற்சி

I. கோடிட்ட இடத்தை நிரப்புக.

1. ஒரு எழுத்து வகையின் அளவு ____________ ல் அளக்கப்படுகிறது.

   ஒரு அங்குலத்துக்கு ____________ புள்ளிகள் உள்ளன.

2. ஒரு ஆவணத்தை வண்ணத்தில் தாள்படி (hardcopy) எடுப்பதற்கு ____________ அச்சுபொறி தேவைப்படும்.

3. ஒரு ஆவணத்தில் முக்கியப் பகுதிகளைக் குறிப்பதற்கு ____________ பண்புக்கூறு பயன்படுகிறது.

4. தேர்வு செய்யப்பட்ட உரையை நேர்த்தி இசைவு செய்வதற்கு ____________ சாவிச் சேர்மானம் பயன்படுத்தப்படுகிறது.

5. உரையை வடிவூட்டல் செய்வதற்கு பயன்படும் பெயரிடப்பட்ட கொடா நிலைமைகளுக்கு (named set of defaults) ____________ என்று பெயர்.

II. சரியா தவறா என்று கூறு.

1. வரி இடைவெளிகள் எப்பொழுதும் ஒரு முழு எண் அளவில் இருக்க வேண்டும்.
2. உள்தள்ளல் ஒரு ஆவணத்தை எளிதாக படிக்க உதவி செய்கிறது.
3. வரி இடைவெளியை மாற்றினால் சொற்களுக்கு இடையே உள்ள இடைவெளியும் மாறும்.
4. ஒரு பட்டியலில் இருந்து புல்லட் குறியை நீக்குவதற்கு அந்த பட்டியலைத் தேர்வு செய்து Delete சாவியை பயன்படுத்த வேண்டும்.
5. ஒரு குறிப்பிட்ட அமைப்பில் உள்ள எழுத்துக்கள் மற்றும் எண்களின் சேர்க்கையை எழுத்து வகை (font) என்று அழைக்கிறோம்.

III. கீழ்க்கண்ட வினாக்களுக்கு விடையளி

1. வன்வடிவூட்டலுக்கும் மென்வடிவூட்டலுக்கும் உள்ள வேறுபாடு என்ன ?
2. எழுத்து வகையைப் பொறுத்தமட்டில் என்னென்ன வடிவூட்டல் மாறுதல்களைச் செய்ய முடியும் ?
3. என்னென்ன வகையான பத்தி இசைவுகளைச் செய்ய முடியும் ?
4. உரையை உள்தள்ளல் என்றால் என்ன ?
5. புல்லட் குறியிடப்பட்ட, மற்றும் எண் வரிசையிடப்பட்ட பட்டியலை எவ்வாறு உருவாக்கலாம் ?

அதிகாரம் 3

எழுத்துப் பிழை திருத்துதல்

3.1 எழுத்துப் பிழை திருத்துதல்

சொற்செயலி தொகுப்பு மென் பொருட்களைப் பயன்படுத்தி உருவாக்கப்படும் ஆவணங்களில் எழுத்துப் பிழை ஏதும் இல்லாமல் இருக்க வேண்டும். இதன் பொருட்டு ஸ்டார் ஆஃபிஸ் ரைட்டரில் ஒரு அகராதியும் (dictionary) எழுத்துக்களைச் சரிபார்க்கும் நிரலும் (Spell - check program) உள்ளது. ஒரு ஆவணம் தட்டச்சு செய்யப்படும் பொழுதோ அல்லது தயாரித்து முடித்தவுடனோ ஸ்டார் ஆஃபிஸ் ரைட்டர் எழுத்துப் பிழையைச் சுட்டிக் காட்டும்.

3.1.1 தட்டச்சு செய்யும் பொழுது எழுத்துப் பிழையைச் சுட்டிக் காட்டுதல்

ஒரு ஆவணம் தட்டச்சு செய்யப்படும் பொழுதே பிழையாக இருக்கலாம் என்று ஸ்டார் ஆஃபிஸ் ரைட்டருக்கு தோன்றும் சொற்களை சுட்டுக் காட்டும் வசதி ஸ்டார் ஆஃபிஸ் ரைட்டரில் உள்ளது. இந்த வசதியை உள்ளதாகவோ அல்லது இல்லாததாகவோ (ON/ OFF) மாற்றுவதற்கு என்ற பணிக்குறி மீது ஒரு கிளிக் செய்ய வேண்டும். இதற்கு தானியங்கு எழுத்துப் பிழை சோதனைக் (Auto spell check) பணிக்குறி என்று பெயர்.

தானியங்கு எழுத்துப் பிழை சோதனை உள்ளதாக இருக்கும் பொழுது (ON Condition), தட்டச்சு செய்யப்படும் ஒவ்வொரு சொல்லும், ஸ்டார் ஆஃபிஸ் ரைட்டர் அகராதியில் உள்ள சொற்களோடு ஒப்பீடு செய்யப்படும். அகராதியில் உள்ள சொற்களிலிருந்து வேறுபடுகின்ற சொற்கள் சிவப்பு நிற நெளி கோடு கொண்டு அடிக் கோடிடப்படும். இந்தச் சிவப்பு நிறக்கோட்டை நிராகரித்து விட்டுத் தொடர்ந்து தட்டச்சு செய்யலாம். ஆவணத்தைத் தயார் செய்து முடித்தபின் பிழைகளைத் திருத்திக் கொள்ளலாம். பிழை திருத்துவதற்கான இரண்டு வழிமுறைகள் பின்வருமாறு

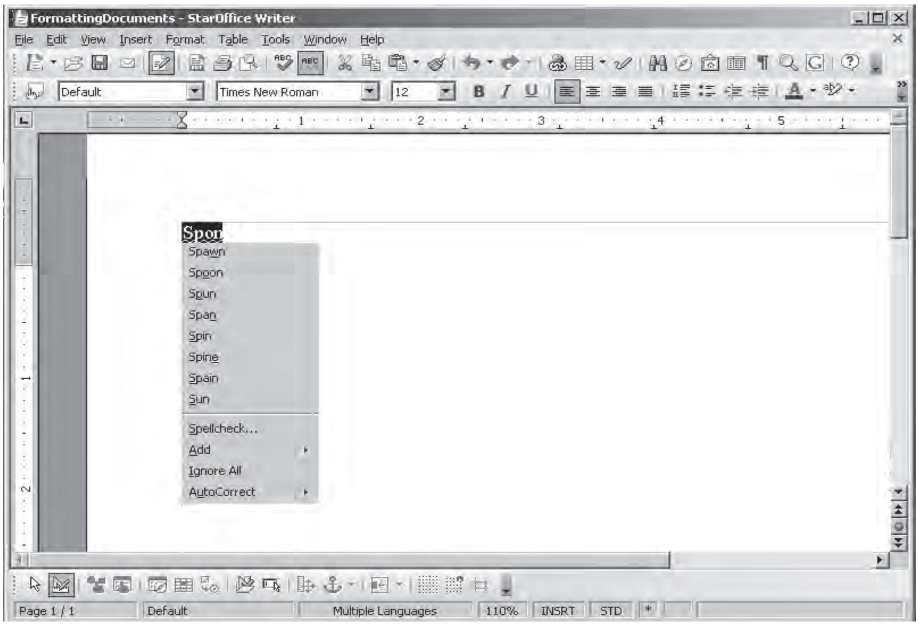
1. Backspace சாவியை அழுத்தி தவறாகத் தட்டச்சு செய்த சொல்லை அழித்துவிட்டு மீண்டும் சரியாகத் தட்டச்சு செய்யலாம்.

2. சரியான எழுத்துக்கள் என்னவென்று தெரியாத பொழுது அந்தச் சொல்லின் மீது சுட்டெலியின் வலது பொத்தானைக் கிளிக் செய்ய வேண்டும். இப்பொழுது ஒரு மேல் மீட்புப் பட்டி திரையில் தோன்றும் இந்தத் திரையில் தோன்றும் சொற்களில் சரியான சொல்லின் மீது கிளிக் செய்ய வேண்டும். இப்பொழுது பிழையான சொல்லுக்குப் பதிலாக சரியான சொல் மாற்றப்பட்டு விடும். உதாரணமாகப் படம் 3.1 இல் Spon என்று தவறாகத் தட்டச்சு செய்யப்பட்ட சொல்லுக்கு ஸ்டார் ஆஃபிஸ் ரைட்டர் தரும் மாற்றான சொற்கள் மேல் மீட்புப் பட்டியில் தரப்பட்டுள்ளன.

3.2 ஆவணத்தைத் தட்டச்சு செய்து முடித்தபின் எழுத்துப் பிழை சரி செய்தல்

கீழ்க்கண்ட வழிமுறைகள் மூலம் ஒரு ஆவணத்தைத் தயார் செய்த பின் எழுத்துப் பிழையை சரி செய்யலாம்.

1. Tools → Spelling → Check என்ற கட்டளையையோ அல்லது என்ற பணிக்குறியையோ கிளிக் செய்ய வேண்டும். ஆவணத்தின் ஒரு குறிப்பிட்ட பகுதியில் எழுத்துப் பிழை சரி செய்ய அதை மட்டும் தேர்வு செய்ய வேண்டும்.



படம் 3.1 மீல் மீட்புப் பட்டி

F7 பொத்தானை அழுத்தியும் எழுத்துப் பிழை திருத்தும் உரையாடல் பெட்டியைப் பெறலாம். F7 பொத்தானை அழுத்தினால் படம் 3.2இல் காட்டப்பட்டுள்ள உரையாடல் பெட்டி தோன்றும்.

படம் 3.2 எழுத்துப் பிழை திருத்தும் உரையாடல் பெட்டி

Not in dictionary பகுதி தவறாகத் தட்டச்சு செய்யப்பட்ட சொல்லைக் காட்டும். Suggestions பட்டியலில் மாற்றுச் சொற்கள் தோன்றும். இப்பொழுது கீழ்க்கண்டவற்றில் ஏதேனும் ஒன்றைச் செய்ய வேண்டும்.

- இந்த முறை தவிர்த்து அடுத்து முறை அந்தச் சொல்லை சுட்டிக் காட்ட Ignore once என்ற பொத்தானைக் கிளிக் செய்ய வேண்டும் அந்தச் சொல் வரும் இடங்களையெல்லாம் சுட்டிக் காட்டாமல் இருக்க Ignore All பொத்தானைக் கிளிக் செய்ய வேண்டும்.

- அந்தச் சொல்லுக்குப் பதிலாக பட்டியலில் உள்ள ஒரு சொல்லை மாற்றியிடுவதற்கு மாற்றாக வர வேண்டிய சொல்லின் மீது ஒருமுறை கிளிக் செய்து பின் Change என்ற பொத்தானைக் கிளிக் செய்ய வேண்டும். அந்தத் தவறான சொல்லுக்குப் பதிலாக வரும் எல்லா இடங்களிலும் மாற்றுச் சொல்லையிடுவதற்கு Change All என்ற பொத்தானைக் கிளிக் செய்ய வேண்டும்.

- Suggestions பட்டியலில் சரியாக மாற்றுச் சொல் இல்லாத பொழுது Not in dictionary பகுதியைப் பயன்படுத்தி சரியான எழுத்துக்களைத் தட்டச்சு செய்ய வேண்டும். Add பொத்தான் கிளிக் செய்தால் இந்தப் புதிய சொல் ஸ்டார் ஆஃபிஸ் ரைட்டரில் சேர்க்கப்படும்.

செய்து பார்

1. Exercise 1 என்ற ஆவணத்தைத் திறக்கவும். தானியங்கு எழுத்துப் பிழை திருத்தி (Auto spell check) வசதியை உள்ளதாக (ON) மாற்றவும். கீழ்க்கண்ட பத்தியை ஏற்கனவே உள்ள உரையோடு சேர்க்கவும், தட்டச்சு செய்யும் பொழுதே வேண்டுமென்றே சில சொற்களைத் தவறாகத் தட்டச்சு செய்யவும். இப்பொழுது தானியங்கு எழுத்துப் பிழை திருத்தி எவ்வாறு வேலை செய்கிறது என்று கவனிக்கவும்.

   It is more than likely that many of today's olympic disciplines are sophisticated versions of the games of strength and speed that flourished in ancient India and Greece, chess, Wresting, Polo, arahergy and hockey(possibly a fall-out from plol) are some of the games believed to have originated in India.

2. இப்பொழுது தானியங்கு எழுத்துப் பிழை திருத்தியை இல்லாததாக (OFF) மாற்றி மேலும் சில தவறான சொற்களைக் கொண்ட பத்தியைத் தட்டச்சு செய்யவும். பின்னர் எழுத்துப் பிழை திருத்தும் உரையாடல் பெட்டி மூலம் அத்தவறுகளைச் சரி செய்யவும்.

3.3 தானியங்கு சரி செய்யும் தேர்வு (Auto Correct Option)

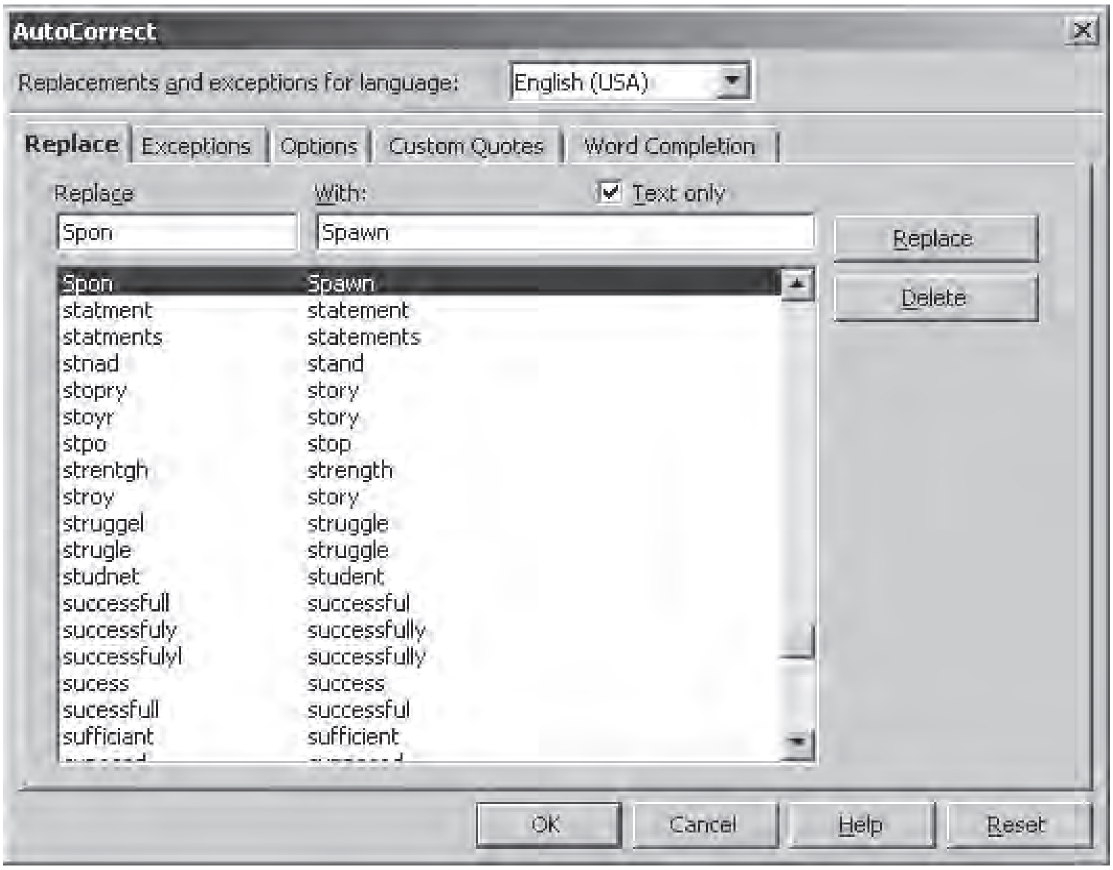
சில வார்த்தைகளை நெளிகோட்டினால் அடிக்கோடிடுவது மட்டுமல்லாமல், ஸ்டார் ஆஃபிஸ் ரைட்டர் தானாகவே சில எழுத்துப் பிழைகளைச் சரி செய்யும். உதாரணமாக “teh” என்று ஒரு சொல்லைத் தானாகவே “the” என்று மாற்றிக் கொள்ளும். இவ்வாறு ஸ்டார் ஆஃபிஸ் ரைட்டர் தானகவே பிழை திருத்தும் சொற்களின் பட்டியலோடு வேறு சொற்களையும் சேர்த்துக் கொள்ளலாம். எடுத்துக்காட்டாக ஒரு பயனர் colour என்ற சொல்லை அடிக்கடி color என்று தவறாகத் தட்டச்சு செய்வாறேயானால் தானியங்கு சரி செய்யும் தேர்வில் இந்தச் சொல்லைச் சேர்த்துக் கொள்ளலாம். அதன் பின்னர் ஒவ்வொரு முறை color என்று சரியான சொல் மாற்றியிடப்படும்.

3.4 தானியங்கு சரி செய்யும் (Auto Correct) பட்டியலில் ஒரு சொல்லைச் சேர்த்தல்

ஒரு தவறான சொல்லும் அதற்குத் சரியான சொல்லையும் தானியங்கு சரி செய்யும் பட்டியலில் சேர்ப்பதற்கான வழிமுறைகள் பின்வருமாறு

Tools → Autocorrect → Auto Format என்ற கட்டளையைத் தேர்வு செய்ய வேண்டும்.

இப்பொழுது படம் 3.3 இல் காட்டப்பட்டுள்ள உரையாடல் பெட்டி தோன்றும். இந்தப் பெட்டியில் Replace என்ற தொகுதியைத் தேர்வு செய்ய வேண்டும். இப்பொழுது Replace என்ற நெடுவரிசையில் மாற்றிடப்பட வேண்டிய சொல்லையும் With என்ற நெடுவரிசையில் மாற்றுச் சொல்லையும் தட்டச்சு செய்ய வேண்டும். பின் OK பொத்தானைக் கிளிக் செய்ய வேண்டும்.



படம் 3.3 தானியங்கு சரி செய்யும் உரையாடல் பெட்டி

இதன் பின்னர் Replace நெடுவரிசையில் உள்ள சொல்லைத் தட்டச்சு செய்தால் ஸ்டார் ஆஃபிஸ் ரைட்டர் தானாகவே With என்ற நெடுவரிசையில் உள்ள சொல்லைக் கொண்டு மாற்றியிடும்.

செய்து பார்

1. கீழ்க்கண்ட சொற்களைத் தட்டச்சு செய்து ஸ்டார் ஆஃபிஸ் ரைட்டரின் தானியங்கு சரி செய்யும் (Auto correct) வசதி எப்படி அதைத் தானாகவே சரி செய்கிறது என்று கவனிக்கவும்.

a) adn

b) actualyl

c) afterthe

d) cna

e) claer

f) teh

g) tihs

h) thta

i) tothe

j) tyhe

2. தானியங்கு சரி செய்யும் உரையாடல் பெட்டியை திறந்து அதில் Replace உரைப்பெட்டியில் Organization என்ற சொல்லையும், With என்ற உரைப்பெட்டியில் Organisation என்ற சொல்லையும் தட்டச்சு செய்யவும். பிறகு OK பொத்தானை அழுத்தவும். இப்பொழுது Organisation என்ற சொல்லைத் தட்டச்சு செய்து என்ன நடக்கிறது என்று கவனிக்கவும்.

சுருக்கம்

- எழுத்துப் பிழைகளை தட்டச்சு செய்யும் பொழுதோ அல்லது தட்டச்சு செய்து முடித்த பின்னரோ சரி செய்து கொள்ள முடியும்.

- தானியங்கு எழுத்துப் பிழை திருத்தும் வசதி மூலம் சில சொற்களில் உள்ள பிழைகளைத் தானாகத் திருத்த முடியும்.

- தானியங்கு சரி செய்யும் (Auto correct) வசதி மூலம் பொதுவாகத் தவறாகத் தட்டச்சு செய்யப்படும் சொற்களை மாற்றி சரியான சொல்லை இடுவதற்குப் பயன்படும்.

பயிற்சி

I. கோடிட்ட இடங்களை நிரப்புக

1. ________________ சாவியைப் பயன்படுத்தி சொற்களின் எழுத்துப் பிழையை சரி செய்யும் உரையாடல் பெட்டியைப் பெறலாம்.

2. ________________ பொத்தானை அழுத்தி முதல்முறை ஒரு சொல்லின் எழுத்துப் பிழையைச் சரி செய்வதைத் தவிர்க்கலாம்.

3. ________________ என்ற தேர்வின் மூலம் தானாகவே தவறான எழுத்துள்ள சொல்லுக்கு பதிலாக சரியான சொல்லை இடலாம்.

II. சரியா தவறா என்று கூறு

1. ஒரு சொல்லுக்குச் சரியான எழுத்துக்கள் தெரியாத பொழுது ஸ்டார் ஆஃபிஸ் ரைட்டர் சரியான எழுத்துக்கள் கொண்ட சற்றேறக்குறைய அதைப் போன்ற சொற்களின் மூலம் பயனருக்கு உதவி செய்கிறது.

2. ஸ்டார் ஆஃபிஸ் ரைட்டர் எழுத்துப்பிழை கொண்ட சொல்லின் அடியில் பச்சைக் கலர் நெளிக்கோடு ஒன்றை இடும்.

3. ஸ்டார் ஆஃபிஸ் ரைட்டர் பொதுவாகத் தவறுதலாகத் தட்டச்சு செய்யப்படும் சொற்களைத் தானாகவே சரி செய்து கொள்ளும்.

III. கீழ்க்கண்ட வினாக்களுக்கு விடையளி

1. தானியங்கு எழுத்துப்பிழை திருத்துதல் என்றால் என்ன ?

2. ஒரு ஆவணம் முழுமையாகத் தட்டச்சு செய்யப்பட்டபின் அதில் எவ்வாறு எழுத்துப் பிழையை கண்டு பிடிக்கலாம் ?

3. ஸ்டார் ஆஃபிஸ் ரைட்டரின் தானியங்கு சரி செய்யும் பட்டியலில் ஒரு சொல்லை எவ்வாறு சேர்க்கலாம் ?

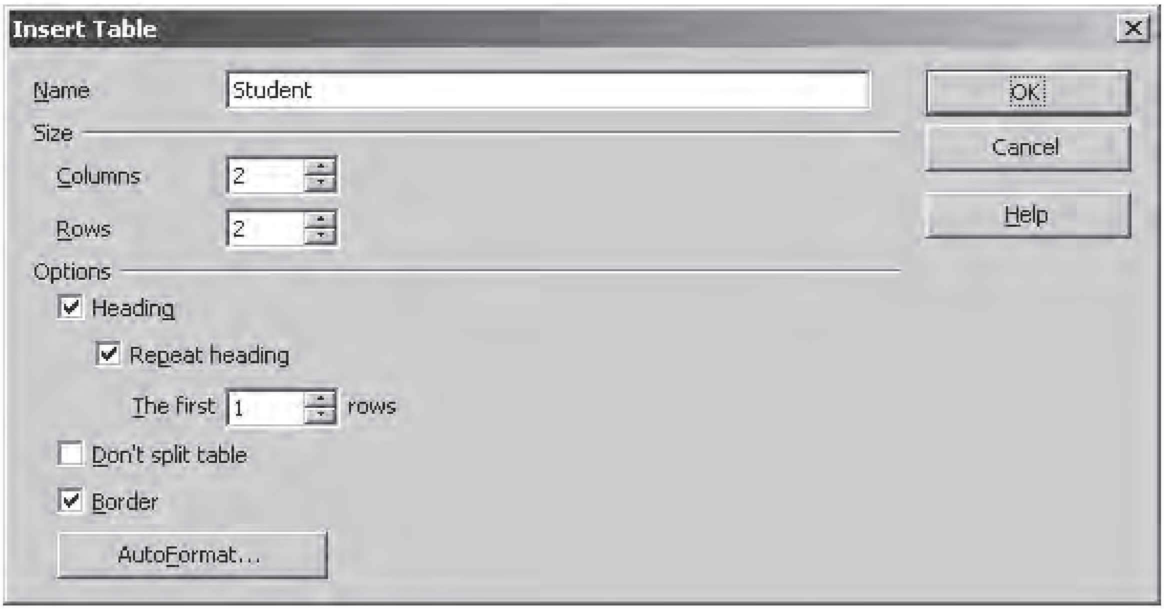
அதிகாரம் 4

அட்டவணையில் வேலை செய்தல்

4.1 ஒரு எளிய அட்டவணையைச் செய்தல்

கீழ்க்காணும் வழிகளைப் பயன்படுத்தி ஒரு எளிய, கொடா நிலை தற்கோள்பாணி (default - style) அட்டவணையைச் செய்யலாம்.

1. தலைமைப் பட்டியிலிருந்து Insert → Table பொத்தான்களைத் தேர்வு செய்து click செய்தவுடன் படம் 4.1 இல் காட்டியுள்ள அட்டவணை நுழைப்பு உரையாடல் பெட்டி (Insert Table dialog box) தோன்றும்.



படம் 4.1 அட்டவணை நுழைப்பு உரையாடல் பெட்டி

2. **Name** என்ற உரைப் பெட்டியில் அட்டவணையின் பெயரைக் கொடுக்க வேண்டும். **Column** மற்றும் **Rows** என்ற உரைப்பெட்டியில் உள்ள சுழல் அம்புக்குறிகளைக் கொண்டு நெடுவரிசை**(Column)** மற்றும் வரிசைகளின் **(Row)** எண்ணிக்கையைக் கொடுக்க வேண்டும். அப்பொழுது ஸ்டார் ரைட்டர் குறிப்பிட்ட நெடுவரிசை மற்றும் வரிசைகளைக் கொண்ட அட்டவணையைக் காட்டும். கொடாநிலையாக ஒவ்வொரு சிற்றரையும் **(cell)** ஒரு எல்லையைக் கொண்டிருக்கும். இந்த எல்லைகளின் தோற்றத்தை அழிக்கவோ அல்லது மாற்றம் செய்யவோ முடியும்.

குறிப்பு: view →Toolbars → Insert பொத்தான்களை தலைமைப் பட்டியிலிருந்து தேர்வு செய்து கிளிக் செய்தவுடன் படம் 4.2 இல் உள்ள மிதவைக் கருவிப்பட்டை தோன்றும். இந்தக் கருவிப்பட்டையில் முதல் பொத்தான் **'Insert Table'** பொத்தானாகும். இதனைக் கிளிக் செய்தவுடன் குறுக்கு நெடுக்கைக் கட்டங்களைக் கொண்ட வலை தோன்றும். சுட்டெலியை வலை மீதுள்ள குறுக்கு மற்றும் நெடுவரிசை மேல் எடுத்துச் சென்று குறுக்கு நெடுக்கை வரிசைகளின் எண்ணிக்கையை தேர்வு செய்து கொள்ளலாம். மாறாக பொத்தான் மீது கிளிக் செய்தால் 2 குறுக்கு வரிசை 2 நெடுவரிசை கொண்ட அட்டவணை தானே தோன்றும்.

படம் 4.2 மிதவைக் கருவிப்பட்டை

4.2 அட்டவணையில் தரவுகளை(data) நுழைத்தல்

அட்டவணையை உருவமைத்தபின் செருகும் புள்ளியானது அட்டவணையின் முதல் சிற்றரையில் இருப்பதைக் காணலாம். ஒரு நெடுவரிசையும், குறுக்கு வரிசையும் சந்திக்கும் இடம் சிற்றரை எனப்படும். ஒரு குறிப்பிட்ட சிற்றரையில் ஒரு எழுத்தைத் தட்டச்சு செய்ய வேண்டுமென்றால் முதலில் அங்கு செருகும் புள்ளியை எடுத்துச் செல்ல வேண்டும். **Tab** சாவியைப் பயன்படுத்தி சிற்றரைகளுக்குள் முன்புறமாகவும் **(forward) Shift + Tab** சாவியைப் பயன்படுத்தி பின்புறமாகவும் **(backward)** நகரலாம்.

ஒரு சிற்றரைக்குள் எவ்வளவு உரைகளை வேண்டுமானாலும் தட்டச்சு செய்யலாம். உரை நீளமாக இருந்தால் அது சிற்றரையின் எல்லையைத் தொட்டவுடன் உரை தானாகவே மடிந்து அடுத்த வரிக்கு வந்து விடும். மேலும் சிற்றரையானது அதற்கேற்ப உயரமாக மாறிவிடும். ஆனால் அதன் அகலத்தில் எந்த மாற்றமும் இருக்காது. சிற்றரையில் உள்ள எழுத்துக்களை முந்திய படங்களில் படித்த வடிவூட்டல் **(formatting)** மற்றும் பதிப்பித்தல் **(editing)** முறைகளைக் கொண்டு மாற்றம் செய்து கொள்ளலாம்.

4.3 வரிசை மற்றும் நெடுவரிசைகளை நீக்குதல் மற்றும் சேர்த்தல்

பெரும்பாலான நேரங்களில் ஒரு அட்டவணையில் தேவைப்படும் நெடுவரிசை மற்றும் வரிசைகளின் எண்ணிக்கையை முன்கூட்டியே தீர்மானிப்பது கடினம். அட்டவணையைத் தயார் செய்தபின் தேவையான நெடுவரிசைகளையோ, அல்லது வரிசைகளையோ சேர்க்கவோ அல்லது நீக்கவோ முடியும்.

4.3.1 வரிசை, நெடுவரிசையைச் சேர்த்தல்

இதற்கு மிகவும் எளிமையான வழி, அட்டவணையின் கடைசி வரிசையில் Tab சாவிப்பொத்தானை அழுத்த வேண்டும். இப்பொழுது ஒரு வரிசை அட்டவணையின் இறுதியில் சேர்க்கப்பட்டிருக்கும். அட்டவணையில் நெடுவரிசையைச் சேர்ப்பதற்கு கீழ்க்கண்ட வழிமுறைகளைக் கையாள வேண்டும்.

1. என்ற பணிக்குறியை கிளிக் செய்தால் செருகும் இடம் உள்ள வரிசைக்கு கீழாக ஒரு வரிசை சேர்க்கப்படும்.

2. என்ற பணிக்குறியை கிளிக் செய்தால் செருகும் இடம் உள்ள நெடுவரிசைக்கு வலது பக்கமாக ஒரு நெடுவரிசை சேர்க்கப்படும்.

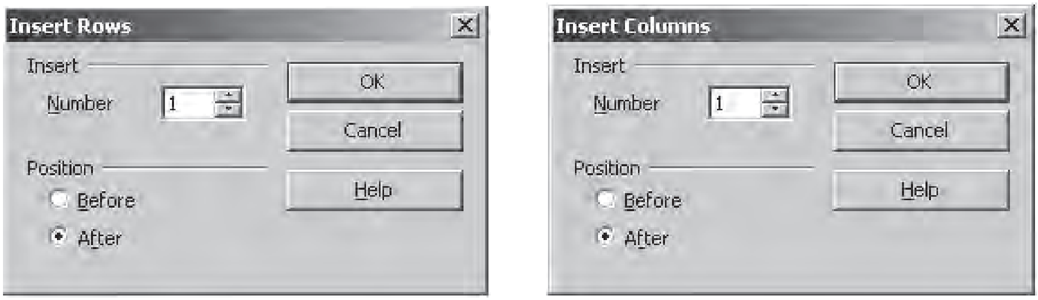
ஒன்றுக்கு மேற்பட்ட வரிசை அல்லது நெடுவரிசைச் சேர்க்க விரும்பினால் **Table→Insert Rows →** அல்லது **Table → Insert → Columns** என்ற கட்டளையைப் பயன்படுத்த வேண்டும். இப்பொழுது படம் **4.3** இல் காட்டப்பட்டுள்ள உரையாடல் பெட்டி தோன்றும். இந்த நெடுவரிசையின் எண்ணிக்கையைக் கொடுக்க வேண்டும். பின் **OK** பொத்தானைக் கிளிக் செய்தால் தேவையான வரிசை மற்றும் நெடுவரிசை அட்டவணையில் சேர்க்கப்படும்.

4.3.2 வரிசை நெடுவரிசையை நீக்குதல்

தேர்வுசெய்யப்பட்ட வரிசை மற்றும் நெடுவரிசையை நீக்குவதற்கு **Table → Delete → Rows** அல்லது **Table → Delete → Columns** கட்டளையைப் பயன்படுத்த வேண்டும்.

4.3.3 முழு அட்டவணையை நீக்குதல்

சுட்டெலி அல்லது விசைப் பலகை மூலம் இழுத்து முழு அட்டவணையை தேர்வு செய்த பின் Delete பொத்தானை அழுத்தினால் அந்த அட்டவணையில் உள்ள பொருளடக்கம் மட்டுமே நீக்கப்படும். அட்டவணை அப்படியே இருக்கும். முழு அட்டவணையை பின்வருமாறு நீக்கலாம்.



படம் 4.3 வரிசை மற்றும் நெடுவரிசை காட்டப்பட்டுள்ள உரையாடல் பெட்டி

1. இப்பொழுது செருகும் புள்ளியை அட்டவணையின் உள்ளே இருத்தி **Table→Select→ Table** என்ற கட்டளையைப் பயன்படுத்தி நீக்க வேண்டிய அட்டவணையை தேர்வு செய்ய வேண்டும்.

2. பிறகு **Table→ Delete → Table** என்ற கட்டளையைப் பயன்படுத்தித் தேர்வு செய்யப்பட்ட அட்டவணையை நீக்கலாம்.

4.4 குறுக்கு வரிசை/ நெடுவரிசையின் அகலத்தை மாற்றுதல்

Insert→Table என்ற கட்டளையின் மூலம் அட்டவணை உருவாக்கப்படும் பொழுது நெடுவரிசையின் அகலமானது, பக்கத்தின் அளவு மற்றும் நெடுவரிசை எண்ணிக்கையைப் பொறுத்து அமையும். ஒவ்வொரு நெடுவரிசையும் ஒரே அகலத்தில் இருக்கும். நெடுவரிசையின் அகலத்தை மாற்ற வேண்டிய தேவை ஏற்பட்டால் அதைப் பின்வருமாறு செய்யலாம்.

- நெடுவரிசையின் அகலத்தை மாற்ற முதலில் சுட்டியை அட்டவணையின் நெடுவரிசையில் வைக்க வேண்டும்.

- பிறகு **Alt** பொத்தானை அழுத்திக் கொண்டு இடது மற்றும் வலது அம்புப் பொத்தான்களைப் பயன்படுத்தி நெடுவரிசையின் அகலத்தை மாற்றிக் கொள்ளலாம்.

- குறுக்கு வரிசையின் அகலத்தை மாற்ற சுட்டியை குறுக்கு வரிசையில் வைக்க வேண்டும். பிறகு **Alt** பொத்தானை அழுத்தியவாறு மேல் மற்றும் கீழ் அம்புப் பொத்தான்களைப் பயன்படுத்தி குறுக்கு வரிசையின் அகலத்தை மாற்றலாம்.

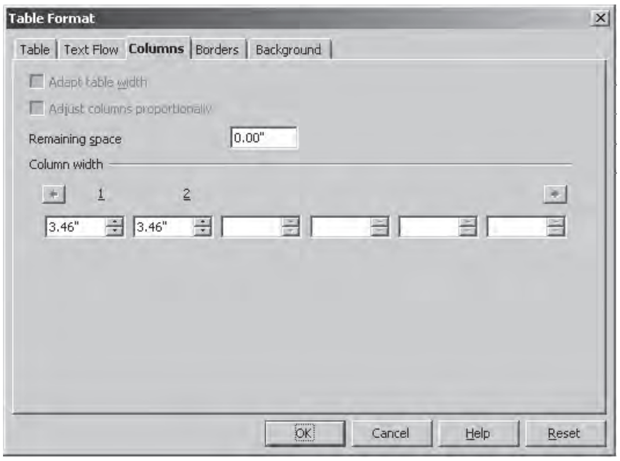
மாறாக **Table → Table Properties** பொத்தான்களைத் தேர்வு செய்தால் **Table Format** உரையாடல் பெட்டி தோன்றும். (படம் **4.4**) பிறகு **Columns tab -** ஐ தேர்வு செய்து நெடுவரிசையின் அகலத்தை அச்சிட்டு **ok** பொத்தானை கிளிக் செய்யவும். நெடுவரிசையின் அகலத்தைத் துல்லியமாகத் தேர்வு செய்ய இந்த வசதி பயன்படும்.

4.4.1 தேர்வு செய்த வரிசை, நெடுவரிசையின் அளவை சமமாக்குதல்

கீழ்க்கண்ட வழிகளில் மூலம் தேர்வு செய்யப்பட்ட வரிசை, நெடுவரிசையின் அளவை ஒரே அளவுள்ளதாக மாற்ற முடியும்.

1. விருப்பமான வரிசை, மற்றும் நெடுவரிசையைத் தேர்வு செய்ய வேண்டும். அனைத்து வரிசை, மற்றும் நெடுவரிசையின் அளவை மாற்ற முழு அட்டவணையையும் தேர்வு செய்ய வேண்டும்.

2. நெடுவரிசையின் அளவை சமமானதாக மாற்ற, சுட்டெலியின் வலது பொத்தானைக் கிளிக் செய்து கிடைக்கும் மேல் மீட்புப் பட்டியில் **Column → Space Equally** என்பதைத் தேர்வு செய்ய வேண்டும். வரிசையின் அளவை மாற்றுவதற்கும் இதே முறையைக் கையாண்டு **Row→Space Equally** என்பதைத் தேர்வு செய்ய வேண்டும்.



படம் 4.4 அட்டவணை வடிவூட்டல் உரையாடல் பெட்டி

4.5 அட்டவணை வடிவூட்டல் கருவிப்பட்டை

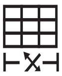
ஒரு அட்டவணையின் உள்ளே சுட்டெலியின் மூலம் ஒரு முறை கிளிக் செய்தால் அட்டவணை தொடர்பான பல பணிகளைச் செய்வதற்கான கருவிப்பட்டை ஒன்று திரையின் மேல் பாகத்தில் தோன்றும். இந்த கருவிப் பட்டையில் படம் 4.5 இல் காட்டியவாறு பல பணிக்குறிகள் இருக்கும். இவற்றின் மூலம் அட்டவணையில் பல்வேறு மாற்றங்களை உருவாக்க முடியும்.

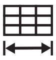
படம் 4.5 அட்டவணை வடிவூட்டல் கருவிப்பட்டை

ஒரு பணிக்குறியின் பணிகள் என்ன என்பது கீழே குறிக்கப்பட்டுள்ளது.

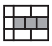
அட்டவணை வடிவூட்டல் கருவிப்பட்டையிலுள்ள ஒவ்வொரு பணிக்குறியின் செயல்பாடுகள் கீழே கொடுக்கப்பட்டுள்ளன.

குறிப்பிட்ட அட்டவணை

 குறிப்பிட்ட அட்டவணை வீதப்படி

 அட்டவணை மாறி

இரண்டு அல்லது அதற்கு மேற்பட்ட சிற்றறைகளை இணைந்து ஒரே சிற்றறையாக மாற்றுவதற்கு

 ஒரு சிற்றறையை இரண்டு அல்லது அதற்கு மேற்பட்ட சிற்றறைகளாக மாற்றுவதற்கு

இந்தப் பணிக்குறியை கிளிக் செய்தால், ஒரு மேல் மீட்புப் பட்டி தோன்றும், இதில் நெடுவரிசையில் அளவு சமம் (Space Column equally), வரிசையின் அளவு சமம் (Space Rows Equally), சரியான வரிசை உயரம் (optimum row height), சரியான நெடுவரிசை அகலம்(optimum row column) போன்ற தேர்வுகள் இருக்கும்.

இதன் மூலம் நடப்பு வரிசையின் கீழ்ப்புறத்தில் ஒரு வரிசையைச் சேர்க்கலாம்.

இதன் மூலம் நடப்பு நெடுவரிசையின் வலது புறத்தில் ஒரு நெடுவரிசையைக் சேர்க்கலாம்.

இதன் மூலம் நடப்பு வரிசையை அழிக்கலாம்.

இதன் மூலம் நடப்பு நெடுவரிசையை அழிக்கலாம்.

இந்த பணிக்குறியைக் கிளிக் செய்தால் ஒரு மிதவை கருவிப்பட்டை தோன்றும். இதன் மூலம் அட்டவணைக்குத் தேவையான எல்லைகளின் (border) வகையைத் தேர்வு செய்ய முடியும்.

இதன் மூலம் அட்டவணையின் எல்லைகளுக்குப் பயன்படுத்தும் கோட்டின் (line) வகையைத் தேர்வு செய்யலாம்.

அட்டவணையின் பின்புறம் (background) என்ன நிறமாக இருக்க வேண்டுமோ அதை இதன் மூலம் கொடுக்க முடியும்.

செய்து பார்

1. மூன்று வரிசை மற்றும் ஆறு நெடுவரிசை கொண்ட ஒரு எளிய அட்டவணையைத் தயாரித்து அதில் கீழ்க்கண்ட தரவுகளை இடவும்.

1	Sunil	100	98	99	100
2	Arvind	99	96	97	100
3	Subish	100	90	100	95

2. மேலே தயாரிக்கப்பட்ட அட்டவணையின் மேல் பாகத்தில் ஒரு வரிசையை நுழைத்து அதில் கீழ்க்கண்ட தரவுகளை சேர்க்கவும்.

Si. No Name Maths Physics Chemistry Computer Science

3. அட்டவணையில் மேலும் இரண்டு நெடுவரிசையை, **Name** மற்றும் **Maths** நெடுவரிசைக்கு இடையில் சேர்த்து அதற்கு **Tamil**, **English** என்று தலைப்பிடவும். அதில் ஒவ்வொரு மாணவருக்கு எதிராகவும் உனக்கு விருப்பமான மதிப்பெண்களை தட்டச்சு செய்யவும். மேலும் கடைசி வரிசைக்கு முன்பாக இரண்டு வரிசையை சேர்த்து அதில் மேலும் இரண்டு மாணவர்களின் மதிப்பெண்களைச் சேர்க்கவும்.

4. அட்டவணையின் ஓரங்கள், கோடுகள் மற்றும் பின்புற வண்ணத்தை மாற்றி அமைக்கவும்.

சுருக்கம்

- ஆவணத்தின் எந்த ஒரு இடத்திலும் அட்டவணையை எளிதாக நுழைக்க முடியும்
- ஒரு அட்டவணையின் தொடக்கத்திலோ இடையிலோ அல்லது இறுதியிலோ வரிசை மற்றும் நெடுவரிசைகளை சேர்க்கவோ நீக்கவோ முடியும்.
- வரிசை மற்றும் நெடுவரிசையின் உயரம், அகலத்தை ஒரு குறிப்பிட்ட சரியான அளவுக்கு எளிதாக மாற்றிக் கொள்ளலாம்.
- அட்டவணை வடிவூட்டல் கருவிப்பட்டையில் உள்ள பல பணிக்குறிகளைக் கொண்டு அட்டவணை தொடர்பான செயல்களை எளிதாகச் செய்யலாம்.

பயிற்சி

I. கோடிட்ட இடத்தை நிரப்புக

1. குறிப்பிட்ட ________________ மற்றும் ___________________ எண்ணிக்கையைக் கொண்ட கட்டத்தை அட்டவணை என்று சொல்கிறோம்.
2. ஒரு அட்டவணையின் சிற்றறைகளுக்குள் பின்புறமாக நகர்வதற்கு ______________ பொத்தான் சேர்மானம் பயன்படுகிறது.
3. _______________ உரையாடல் பெட்டியின் துணை கொண்டு நெடுவரிசைகளின் அகலத்தை ஒரே அளவாக மாற்ற முடியும்.
4. அட்டவணையின் ஓரங்களை மாற்றுவதற்கு ________________________ கருவிப்பட்டை துணை செய்கிறது.
5. ஒரு நெடுவரிசையை அட்டவணையில் இருந்து நீக்குவதற்கு பயன்படும் கட்டளைகள் __________________, __________________, __________________ ஆகும்.

II. சரியா தவறா என்று கூறு.

1. ஒவ்வொரு அட்டவணையும் ஒரு குறிப்பிட்ட பெயரைக்கொண்டு தான் அடையாளம் காட்ட முடியும்.
2. ஒரு அட்டவணையில் வரிசையையோ அல்லது நெடுவரிசையையோ முதலிலோ அல்லது இறுதியிலோ தான் சேர்க்க முடியும்.

3. ஒரு அட்டவணையை முழுவதுமாக நீக்குவதற்கு அதனை தேர்வு செய்து **Delete** சாவியை அழுத்த வேண்டும்.

4. அட்டவணையை வடிவூட்டல் செய்வதற்கென்று தனியாக ஒரு கருவிப்பட்டை உள்ளது.

5. அட்டவணையின் எல்லா நெடுவரிசைகளும் ஒரே அகலத்தில் இருக்கும்.

III. கீழ்க்கண்ட வினாக்களுக்கு விடையளி

1. ஒரு ஆவணத்தில் எவ்வாறு ஒரு அட்டவணையை உருவாக்கலாம் ?

2. ஒரு அட்டவணையில் தேவையான வரிசை மற்றும் நெடுவரிசையை எப்படிச் சேர்க்கலாம் ?

3. ஒரு அட்டவணையில் நெடுவரிசையின் அகலத்தை எவ்வாறு மாற்றலாம். ?

4. அட்டவணை வடிவூட்டல் பட்டிப்பட்டையில் உள்ள குறும்படங்களின் வேலைகள் என்ன ?

5. தேர்வு செய்யப்பட்ட வரிசை மற்றும் நெடுவரிசையின் அகலத்தை ஒரே அளவுள்ளதாக எவ்வாறு மாற்றலாம் ?

அதிகாரம் 5

பக்க வடிவமைப்பு

5.1 ஓரத்தை (Margins) மாற்றுதல்

அச்சிடப்பட்ட பகுதிக்கும் பக்கத்தின் இடது பக்க விளிம்புக்கும் இடையில் உள்ள இடத்துக்கு ஓரம் **(Margin)** என்று பெயர். எடுத்துக்காட்டாக மேல் ஓரம் மிகுதியாக இருந்தால் ஆவணத்தின் மேல் பகுதியில் வெள்ளைப் பகுதி அதிகமாகவும், குறைவாக இருந்தால் வெள்ளைப் பகுதி குறைவாகவும் இருக்கும்.

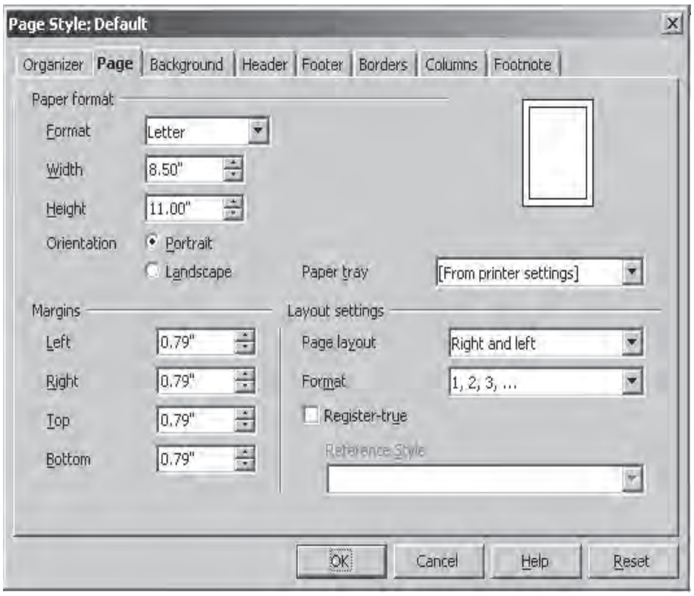
ஸ்டார் ஆஃபிஸ் ரைட்டர் **(Star Office Writer)** ஆவணத்தின் கொடாநிலை ஓரம் மேல் பாகத்திலும் கீழ்பாகத்திலும் 1 அங்குலமும் வலது மற்றும் இடது பாகத்தில் 1.25 அங்குலமும் ஆகும். பெரும்பாலான ஆவணங்களுக்கு இது போதுமானதாக இருக்கும். தேவைப்படும் பொழுது இந்த நான்கு அளவுகளையும் மாற்றி அமைத்துக் கொள்ளலாம்.

கீழ்க்காணும் ஏதேனும் ஒரு முறையைப் பயன்படுத்தி ஓரத்தின் அளவை மாற்றலாம்:

5.1.1 பக்க அமைப்பு உரையாடல் பெட்டியைப் (Page Style Dialog Box) பயன்படுத்துதல்

ஒவ்வொரு ஓரத்துக்கும் சரியான அளவு தெரியும் பொழுது பக்க அமைப்பு உரையாடல் பெட்டியைப் பயன்படுத்தலாம். கீழ்க்காணும் செயல்பாட்டைப் பயன்படுத்தவும்:

- **Format → Page** கட்டளையைத் தேர்வு செய்ய வேண்டும். உடனே படம் 5.1 இல் காட்டியுள்ள உரையாடல் பெட்டி தோன்றும்.
- தேவையானால் **Page Tab –** ஐ கிளிக் செய்யவும்.
- ஓர அளவுப் பகுதியில் உள்ள சுழல் பெட்டியில் புதிய அளவை அச்சிடவும். அல்லது சுழல் அம்புகளைப் பயன்படுத்தி அளவை மாற்றவும்.
- **OK** பொத்தானை கிளிக் செய்யவும்.



படம் 5.1 பக்க அமைப்பு உரையாடல் பெட்டி

File பட்டியில் உள்ள **Page Preview** பொத்தானைப் பயன்படுத்தி செய்த மாற்றங்களைப் பார்க்கலாம். இந்த வசதி அச்சிட்டால் ஆவணம் எப்படி இருக்கும் என்பதை அறியவும், செய்த மாற்றங்கள் ஆவண வடிவை எப்படி மாற்றியுள்ளது என்பதை அறியவும் உதவும்.

5.1.2 ரூலரின் (Ruler) துணை கொண்டு ஓரங்களை மாற்றுதல்

ஆவண வடிவின் ஓர அளவுகள் சரியாகத் தெரியவில்லையெனில் **View** பட்டியில் உள்ள **Ruler** வசதியைப் பயன்படுத்தி ஓரத்தின் அளவுகளை மாற்றிக் கொள்ளலாம். அதற்கு கீழ்க்காணும் முறையைப் பயன்படுத்துக.

- **Ruler** திரையில் தோன்றாவிட்டால் **View→Ruler** பொத்தானை அழுத்த வேண்டும்.
- **Ruler** –இன் சாம்பல்நிறப் பகுதி ஓர அளவின் மேல் பகுதியைக் குறிக்கிறது. சுட்டெலியின் சுட்டியை சாம்பல் நிறப்பகுதிக்கும் வெள்ளை நிறப்பகுதிக்கும் இடையில் எடுத்துச் செல்ல வேண்டும்.
- சுட்டி சரியான இடத்தில் இருந்தால் அந்தச் சுட்டி இருதலை கொண்ட அம்புக் குறிபோல் காட்சியளிக்கும்.
- இப்பொழுது ஓர வழிகாட்டியை **(margin guide)** புதிய இடத்துக்கு நகர்த்த வேண்டும்.
- **Close** பொத்தானைக் கிளிக் செய்தால் இயல்பான பார்வை **(Normal View)** கிடைக்கும்.

5.2 பக்கத்தின் அமைவுகள் (Page Orientation)

பக்கத்தின் நீளம் அகலத்தைவிட அதிகமாக இருந்தால் அது போர்ட்ரைட் **(Portrait)** அமைவு எனப்படும். அதே சமயம் அகலம் நீளத்தைவிட அதிகமாக இருந்தால் அது லேண்ட்ஸ்கேப் **(Land Scape)** அமைவு எனப்படும்.

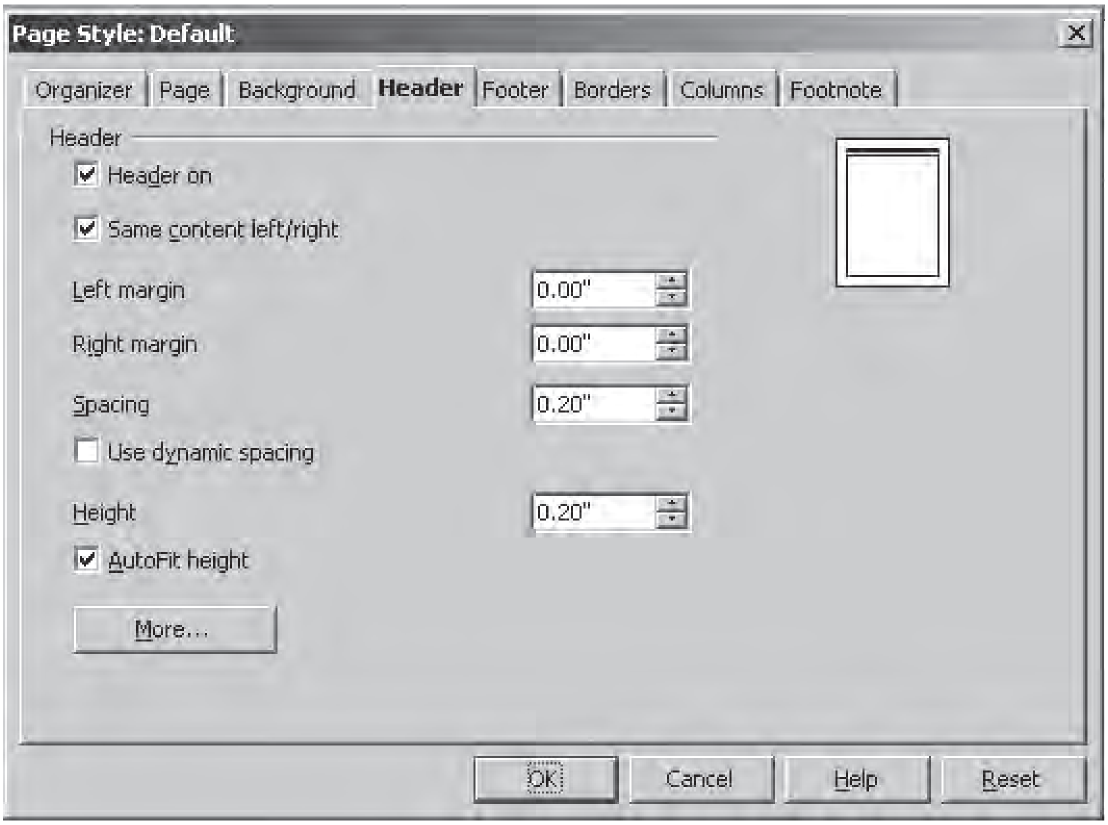
1. **Format → Page** கட்டளையைத் தேர்வு செய்தால் பக்க அமைப்பு உரையாடல் பெட்டி தோன்றும். (படம் 5.1)
2. **Page** தொகுதியைக் கிளிக் செய்து விருப்பங்களைத் தேர்வு செய்யலாம்.
3. தாளின் அளவை மாற்றுவதற்கு **PaperFormat** கீழிறங்கு பட்டியில் வேண்டிய அளவைத் தேர்ந்தெடுக்க வேண்டும். அல்லது சுழல் அம்புக் குறியைப் பயன்படுத்தி உயரம் மற்றும் அகலத்தை மாற்றலாம்.
4. **Orientation** பகுதியில் **Portrait** அல்லது **Land Scape** என்பதைத் தேர்வு செய்ய வேண்டும்.
5. **OK** பொத்தானை கிளிக் செய்ய வேண்டும்.

5.3 தலைப்பு, அடிக்குறிப்புகளை உருவாக்குதல்

ஒன்றுக்கு மேற்பட்ட பக்கங்களைக் கொண்ட ஆவணங்களுக்கு பக்க எண்கள் சேர்த்துக் கொள்ளப்படுகின்றன. பக்க எண்கள் முக்கியமானதாகும். இல்லையெனில் குறிப்பிட்ட பக்கத்தைக் கண்டுபிடிப்பது மிகவும் கடினமாகும். இது தவிர பயனர் மேலும் சில சொற்களை பக்க எண்களோடு குறிப்பிட விரும்பலாம். உதாரணமாக ஆவணத் தலைப்பு அல்லது நூலாசிரியர் **(Author)** பெயரையோ எல்லாப் பக்கங்களிலும் சேர்க்க விரும்பலாம். இதற்காக

ஒவ்வொரு பக்கத்திலும் தனித்தனியாக தட்டச்சு செய்ய வேண்டியதிற்குப் பதிலாக தலைப்பு அல்லது அடிக்குறிப்புகள் பயன்படுத்தப்படுகிறது. தலைப்பு என்பது பக்கத்தின் மேல்பகுதியையும் அடிக்குறிப்பு என்பது பக்கத்தின் கீழ்ப்பகுதியையும் குறிக்கும். பயனர் தன் விருப்பத்திற்கேற்ப அப்பகுதியில் தட்டச்சு செய்து கொள்ளலாம். ஸ்டார் ஆஃபிஸ் ரைட்டர் அப்பகுதியை ஒவ்வொரு பகுதியிலும் தானாகவே சேர்த்துக் கொள்ளும்.

ஒரு தலைப்பை உருவாக்குவதற்கு **Header** என்னும் **Tab** பயன்படுத்தப்படுகிறது. படத்தில் காட்டியுள்ளவாறு வெவ்வேறு விதமான விருப்பத் தேர்வுகளுடன் கூடிய ஒரு திரை தோன்றும். இந்தத் திரையில் தோன்றும் உரையாடல் பெட்டியில் **(dialog box)** உள்ள **Header** என்ற பொத்தானை கிளிக் செய்ய வேண்டும். இதில் நான்கு சுழல் பெட்டிகள் தோன்றும். இந்த சுழல் பெட்டிகள் தலைப்பிற்கும், ஆவணத்திற்கும் உள்ள தூரத்தை மாற்றவும் தலைப்பின் உயரம் மற்றும் இடது ஓரம், வலது ஓரம் ஆகியவற்றை மாற்றுவதற்கும் பயன்படுகிறது.



படம் 5.2 பக்க அமைப்பு உரையாடல் பெட்டி

இதைப் போன்றே செய்முறைகளே ஒரு அடிக்குறிப்பை உருவாக்குவதற்கும் பயன்படுகிறது. இதில் ஒரு வேறுபாடு என்னவென்றால் இதில் **Header Tab –** க்குப் பதிலாக **Footer Tab -** ஐக் கிளிக் செய்யப்பட வேண்டும்.

ஒரு ஆவணப்பக்கத்தில் பக்க அமைப்பு உரையாடல் பெட்டி **(Page Style Dialog Box)** மூலமாக படம் **5.3 –** இல் காட்டியுள்ளவாறு தலைப்பு மற்றும் அடிக்குறிப்புகளை இப்பொழுது மீதமுள்ள ஆவணங்களிலிருந்து தலைப்பு மற்றும் அடிக்குறிப்புப் பகுதிகள் தனியாக ஒரு சின்னக் கோட்டினால் பிரிக்கப்பட்டிருப்பதைக் காணலாம்.

Header
6.4 Creating a Header and Footer When the documents are longer than one page normally some type of reference numbers are added on the page. For instance, page numbers are a must. In addition, the user may want to include other text that helps the reader identified the document. For example, the document title or the author's name can be included in every page. Rather than typing this
Footer

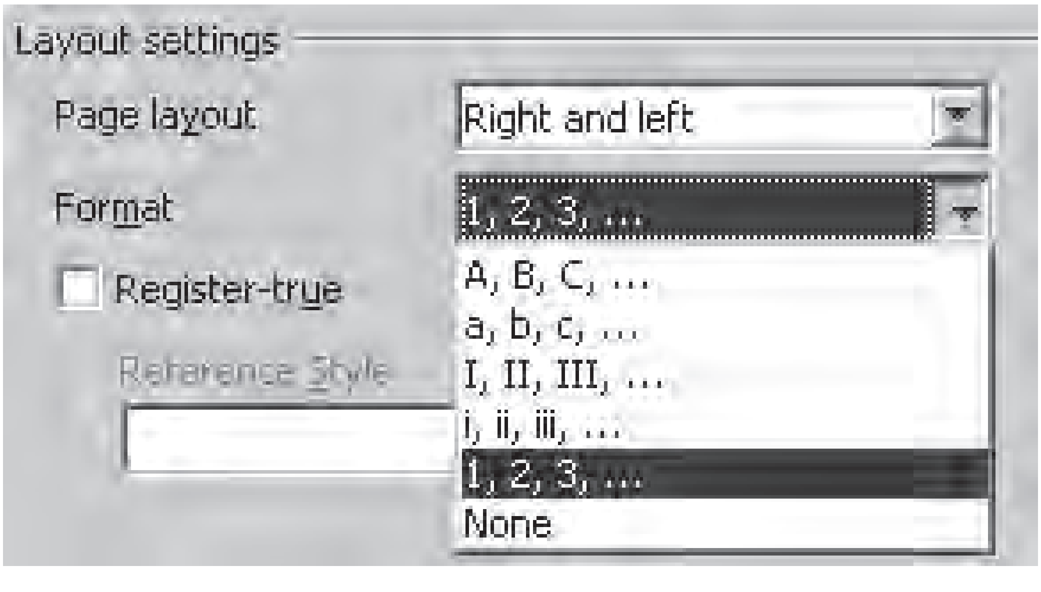
படம் **5.3** தலைப்பு மற்றும் அடிக்குறிப்புப் பகுதிகள் கொண்ட ஒரு ஆவணம்

தலைப்பு மற்றும் அடிக்குறிப்புகளை ஒருமுறை உருவாக்கி விட்டால் அதில் சேர்க்க வேண்டிய உரையைக் குறிப்பிட வேண்டும். அதற்கு கீழ்க்கண்ட வழிமுறைகள் பயன்படுத்தப்படுகின்றன.

1. தலைப்பு அல்லது அடிக்குறிப்புப் பகுதியில் கிளிக் செய்ய வேண்டும்.

2. **Insert → Fields** என்பதைக் கிளிக் செய்தால் ஒரு பட்டியலுடன் கூடிய துணைப்பட்டி தோன்றும். அந்தப் பட்டியலில் ஒன்றைத் தேர்வு செய்தால் அந்த தேர்வு செய்யப்பட்ட உரையானது ஆவணத்தின் எல்லாப் பக்கங்களிலும் கொடுக்கப்பட்டுவிடும். எடுத்துக்காட்டாக **Insert →Fields → Page Number** என்பதைக் கிளிக் செய்தால் ஒவ்வொரு பக்கத்திலும் பக்க எண்கள் புகுத்தப்பட்டுவிடும். தலைப்பு மற்றும் அடிக்குறிப்புப்பகுதியில் இயல்பான உரைகளையும் புகுத்தலாம். அவ்வாறு செய்ய வேண்டுமென்றால் நமக்கு விருப்பமான உரையை வேண்டிய இடத்தில் தட்டச்சு செய்ய வேண்டும்.

பக்க எண்கள் கொடுக்கப்பட்டவுடன் படம் **5.1–**இல் உள்ள பக்க அமைப்பு உரையாடல் பெட்டியின் துணை கொண்டு பக்க எண்களின் வடிவூட்டலை **(Format)** மாற்றி அமைக்கலாம். **(Format)** என்ற சேர்மானப் பெட்டியில் கிளிக் செய்தால் பல வகையான பக்க அமைப்புகள் படம் **5.4 –** இல் காட்டியுள்ளவாறு தோன்றும். அதில் வேண்டிய ஒன்றைத் தேர்வு செய்து கொள்ளலாம்.



படம் 5.4 பக்க எண் சேர்மானப்பெட்டி

செய்து பார்

1. Exerics 1 என்ற ஆவணத்தைத் திறந்து அதன் ஓரங்களை மாற்றி அந்த மாற்றங்களைப் பார்க்கவும்.

2. ஓரங்களை ஏற்கனவே இருந்தவாறு மாற்றவும். Print Preview–ஐப் பயன்படுத்தி அது சரியாக இருக்கிறதா என்று பார்த்துக் கொள்ளலாம்.

3. காட்சிப் பெட்டியில் (View Menu) உள்ள ரூலரைத் தேர்வு செய்து அதன் முலம் ஓரங்களை மாற்றி ஏற்படும் மாற்றங்களைப் பார்க்கவும்.

4. பக்க அமைப்புகளை மாற்றி அதன் முடிவுகளைப் பார்க்கவும்.

5. ஒரு ஆவணத்தில் தலைப்பு மற்றும் அடிக்குறிப்புகளை உருவாக்கவும். தலைப்புப் பகுதியில் தேதி (Date) மற்றும் ஆவணத் தலைப்பை தட்டச்சு செய்யவும். அடிக்குறிப்புப் பகுதியில் பக்க எண்ணை புகுத்தவும்.

சுருக்கம்

- ஒரு குறிப்பிட்ட பக்கத்திற்கு ஓரங்களை பக்க அமைப்பு உரைப்பெட்டி மூலம் துல்லியமான அளவுகளில் அமைக்கலாம். அல்லது ரூலரைப் பயன்படுத்தி தோராயமாகவும் அமைக்கலாம்.

- இரண்டு விதமான பக்க அமைவுகள் உள்ளன. அவை portrait மற்றும் LandScape பக்க அமைவுகள் ஆகும்.

- மாறுபட்ட ஸ்டைலைக் கொண்ட பக்க எண்களை ஸ்டார் ஆஃபிஸ் ரைட்டர் ஆவணத்தில் சேர்க்க முடியும்.

- சில பயனுள்ள குறிப்புகளை பக்கத்தின் மேல் ஓரத்திலும் கீழ் ஓரத்திலும் தருவதற்கு தலைப்பு மற்றும் அடிக்குறிப்புகள் உதவுகின்றன.

பயிற்சி

I. கோடிட்ட இடத்தை நிரப்புக

1. ஸ்டார் ஆஃபிஸ் ரைட்டர் (Star Office Writer) ஆவணத்தின் கொடாநிலை ஓரம் மேல் பாகத்திலும், கீழ் பாகத்திலும் ____________________ அளவும் வலது மற்றும் இடது பாகத்தில் ____________________ அளவும் இருக்கும்.

2. ______________, _________________ ஆகியவை திரையின் மீது பக்கங்களின் வடிவத்தை மாற்றுவதற்கு உதவுகிறது.

3. _______________ உரைப்பெட்டி பக்கத்தின் அகலத்தை அமைப்பதற்கு பயன்படுகிறது.

II. சரியா தவறா என்று கூறுக

1. தலைப்பு என்பது பொதுவாக பக்க எண்களைக் கொண்டிருக்கும்.
2. பக்க ஓரங்களின் அளவுகள் துல்லியமாக அளவுகளில் அமைக்கப்பட வேண்டும்.
3. விருப்பமான பக்க அமைவை பக்க உரைப் பெட்டியில் உள்ள சேர்மானப் பெட்டியின் (Combo Box) மூலமாகத் தேர்வு செய்யலாம்.
4. Ruler - ஐப் பயன்படுத்தி ஆவணப் பக்கங்களின் ஓர அளவுகளை மாற்றி அமைக்கலாம்.
5. Format முறையாக்கத்தை LandScape முறையாக்கம் என்றும் கூறலாம்.

III. கீழ்க்கண்ட வினாக்களுக்கு விடையளி

1. பக்க வடிவமைப்பு என்றால் என்ன ?
2. இரண்டு வகையான பக்க அமைவுகள் யாவை ?
3. பக்கங்களின் ஓரங்களை மாற்றுவதற்கு ரூலர் எவ்வாறு உதவுகிறது ?
4. தலைப்பு, அடிக்குறிப்பு என்பதன் பொருள் என்ன ?
5. Ruler - ஐப் பயன்படுத்தி ஆவணப் பக்கங்களின் ஓர அளவுகளை மாற்றி அமைக்கும் முறையை விளக்குக.

அதிகாரம் 6

அட்டவணைச் செயலி (Spread Sheet)

6.1 முன்னுரை

தனியாள் கணிப்பொறி **(PC)** வகை முதன்முதலின் 1975 –ஆம் ஆண்டில் வெளிவந்தது. ஆனால், தொடக்க காலத்தில் இவை, மின்னணுக் கருவிகளை பொழுதுபோக்காகக் கொண்ட சிலருக்கு மட்டுமே ஆர்வமூட்டுவதாக இருந்தன. பயன்மிக்க தனியார் கணிப்பொறிகள் வரத் தொடங்கிய பிறகு, இவை பலரின் கவனத்தை கவரத் தொடங்கின. 1979 –ஆம் ஆண்டு ஆப்பிள் – **II (Apple - II)** என்ற கணிப்பொறிக்காக டான் பிலிக்ளின் **(Dan Bricklin)**, பாப் ஃப்ராங்க்ஸ்டன் **(Bob Frankston)** என்ற அறிவியல் அறிஞர்கள் ‘விசிகால்க்’ **(Viscalc)** என்னும் முதல் மின் அட்டவணைச் செயலி **(Spread Sheet)** யைக் கண்டுபிடித்தனர். அதன்பின் தனியாள் கணிப்பொறிகள் **(Personal Computer)** மக்களிடையே பெரும் வரவேற்பைப் பெற்றன. ‘விசிகால்க்’ அட்டவணைச் செயலியைக் கொண்ட தனியாள் கணிப்பொறிகள், கணக்காயர்கள், புத்தக விற்பனையாளர்கள், மேலாளர்கள், வரவு – செலவுத் திட்டமிடுவோர், புள்ளிவிவரம் கணக்கிடுவோர், எண் கணித ஆய்வுத்தரவுகளை சேகரிப்போர் போன்றவர்களை பெரிதும் கவரத் தொடங்கின. மிகுந்த திறமையுடன் கணிக்கவல்ல இந்த கணிமுறைக் கருவிகளைக் கொண்டு கணிப்பீட்டு நேரத்தைக் குறைக்கவும், மூளையைக் குழப்பக்கூடிய, வாழ்நாளில் முடிக்கவியலாத எண் கணக்குகளை விரைவாகவும் எளிதாகவும் செய்யவும், பிழையின்றி கணக்கிடவும் முடிந்தது. வேறு எந்த வித கண்டுபிடிப்புகளும் தராத வளர்ச்சியை, மின் அட்டவணைச் செயலியின் கண்டுபிடிப்பு, தனியாள் கணிப்பொறிக்குத் தந்து ஒரு பெரும் புரட்சியை உருவாக்கிவிட்டது.

6.1.1. ‘மின் அட்டவணைச் செயலி’ என்றால் என்ன ?

‘மின் அட்டவணைச் செயலி’ என்பது கணிப்பொறியில், நெடுவரிசைகளைக் கொண்ட அட்டவணை அறிக்கைகளை உருவாக்கவும் அவற்றில் மேற்கொண்டு கணக்கீடுகளைச் செய்யவும் பயன்படும் ஒரு கணக்கீட்டுத்தாள் ஆகும்.

அட்டவணைச் செயலி படம் **6.1**–இல் காட்டப்பட்டுள்ளதைப் போல வரிசைகளையும் நெடுவரிசைகளையும் கொண்டு அமைக்கப்பட்டுள்ளது. வரிசைகளும் நெடுவரிசைகளும் குறுக்கிடும் போது நுண்ணறைகள் **(Cell)** உருவாகின்றன. ஒவ்வொரு நுண்ணறையும் வரிசை மற்றும் நெடுவரிசை எண்களால் அறியப்படுகின்றன. நுண்ணறைகளில் எண் **(Number)**, உரை (எழுத்து) **(Text)** அல்லது வாய்பாடு **(Formula)** போன்ற தரவுகளை **(Data)** எழுதிக் கொள்ளலாம்.

நுண்ணறைகளில் வாய்பாடுகளை இட்டு, அந்த வாய்பாடுகளைக் கொண்டு, பிற, நுண்ணறைகளில் உள்ள தரவுகளை சில கணக்கீடுகளுக்கு உட்படுத்தி, முடிவுகளை பதிய நுண்ணறைகளில் இட்டுக் காட்ட இயலும் இந்த செயலாக்கத் திறன்தான் அட்டவணைச் செயலியின் சிறப்பாகும்.

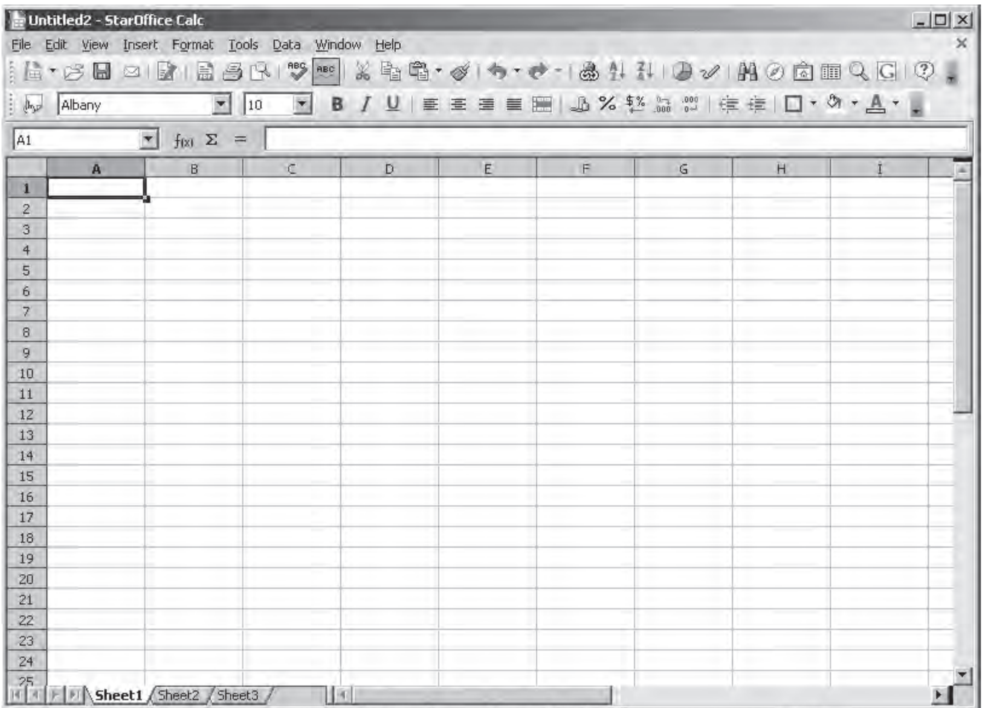
அட்டவணைச் செயலியில் வரிசைகள் **1** முதல் ஒரு குறிப்பிட்ட எண் வரை (‘ஸ்டார் கால்க்’ –இல் **32000** வரை) எண்ணால் குறிக்கப்படுகின்றன. நெடுவரிசைகள் **A** முதல் **Z** வரையும், மேலும், **AA** முதல் **AZ** வரையும் அதற்கு மேலும் எழுத்தால் குறிக்கப்படுகின்றன. வரிசை நெடுவரிசைகளின் எண்ணிக்கைகள் ஒவ்வொரு அட்டவணைச் செயலி தொகுப்பைப் பொருத்து இருக்கும்.

நுண்ணறைகள்**A1, A2….,B1, B2……, AA1, AA2……** என்ற பெயர்களில் அறியப்படும். எடுத்துக்காட்டாக நுண்ணறை **B5** என்பது **B** என்ற நெடுவரிசையும் 5 என்ற வரிசையும் குறுக்கிடும் இடத்தில் உருவாகும் ஊடறை ஆகும்.

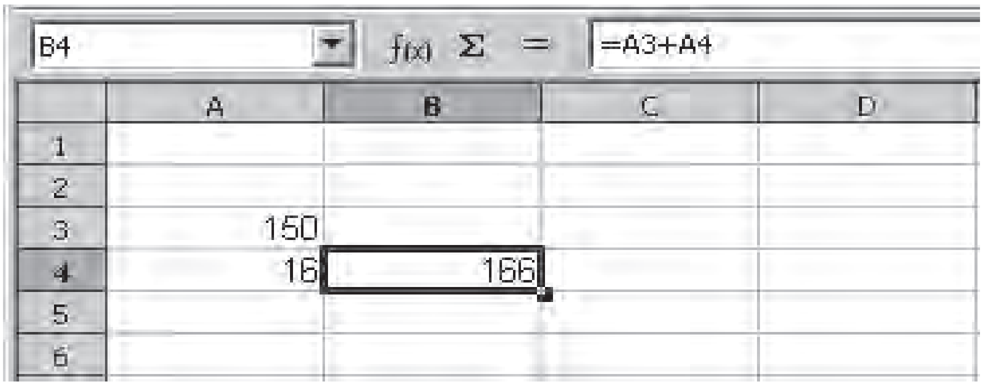
கீழ்க்கண்ட எடுத்துக்காட்டு, அட்டவணைச் செயலியை எப்படி பயன்படுத்துவது என்பதை விளக்குகின்றது.

A3 என்ற நுண்ணறையில் 150 என்ற எண்ணை இட்டுத் தேக்குவதாகக் கொள்வோம். 16 என்ற மற்றொரு எண்ணை வேறு ஒரு நுண்ணறையில் இட்டு தேக்குவோம். பிறகு இந்த இரண்டு எண்களையும் கூட்டி விடையை **B4** என்ற ஒரு புதிய நுண்ணறையில் இட்டு தேக்குவதாகக் கொள்வோம். இதனை அட்டவணைச் செயலியில் எப்படிச் செய்வது என்று பார்ப்போம்.

150 என்ற எண்ணை A3 என்ற நுண்ணறையில் இட, சுட்டி **(Cursor)** ஐ,A3 க் கொண்டு சென்று சுட்டெலி **(Mouse)** யின் இடப்புற பொத்தானை அழுத்தி நுண்ணறை **A3–**ஐ தேர்ந்தெடுத்துக் கொள்ள வேண்டும். பிறகு எண் 150 –ஐ விசைப்பலகை **(Key Board)** மூலம் எழுதி உள்ளிட்டால் நுண்ணறை A3–யில் 150 என்ற எண் தோன்றுவதைக் காணலாம். அதேபோல நுண்ணறை **A4–**இல் 16 என்ற எண்ணை உள்ளிடவும். இப்பொழுது நுண்ணறைகள் **A3–**யிலும் **A4** லும் உள்ள எண்களைக் கூட்டி விடையை நுண்ணறை **B4–**இல் இட்டு தேக்குவதற்கான வழி முறையைக் காண்போம். இதனைச் செய்ய, நுண்ணறை **B4** க்குச் சுட்டியைக் கொண்டு சென்று சுட்டெலியின் இடப்புற பொத்தானை அழுத்தி **B4 -** ஐ தேர்ந்தெடுத்துக் கொள்ளவும். பிறகு = **A3 +A4** அல்லது = **SUM (A3:A4)** என்ற வாய்பாட்டை விசைப் பலகை மூலம் எழுதி உள்ளிடவும். பிறகு வாய்பாடு, வாய்பாட்டுப்பட்டையின் உள்ளீட்டுக் கோட்டில் தோன்றும். இப்பொழுது, **Enter** பொத்தானை அழுத்தவும். உடனே, **A3, A4** நுண்ணறைகளில் உள்ள எண்களின் கூட்டுத் தொகையான 166 என்ற எண் படம் 6.2 இல் காட்டியுள்ளதைப் போல நுண்ணறை **B4–**இல் தோன்றுவதைக் காணலாம்.

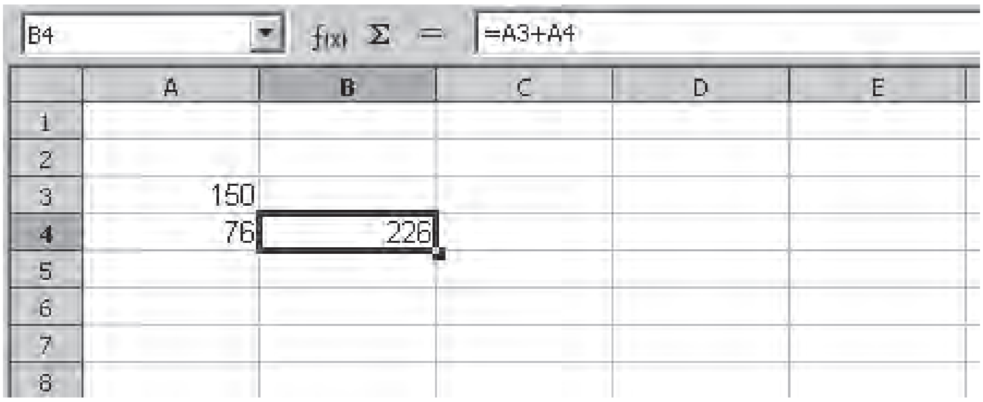


படம் 6.1 அட்டவணைச் செயலியின் வெற்று அட்டவணை



படம் 6.2 அட்டவணைச் செயலியில் வாய்பாடு மூலம் இரண்டு எண்களின் கூட்டுத் தொகையைக் காட்டும் திரை

அட்டவணைச் செயலி, கட்டளை நிரலால் செயல்படும் கணிப்பான் (Programmable Calculator) இணைக்கப்பட்ட நுண்ணறைகளைக் கொண்ட வலையமைப்பைப் போன்றதாகும். அட்டவணைச் செயலி கணக்கீடுகளை கண்ணிமைக்கும் நேரத்தில் செய்யக்கூடியவை. அட்டவணைச் செயலியில் எவ்வகை எண்களைப் பதித்தாலும், எத்தனை வாய்பாடுகளைக் கொடுத்தாலும் கவலையில்லை, வாய்பாட்டில் உள்ள நுண்ணறைகளின் மதிப்புகளை நாம் மாற்றும்போதெல்லாம், ஒவ்வொரு வாய்பாடும் கணக்கீடுகளை மீண்டும் செய்து புதிய அட்டவணையில் விடையைக் காட்டும். எடுத்துக்காட்டாக, **A4–**இல் உள்ள 16 என்ற எண்ணை 70 ஆக மாற்றுவோமானால், அட்டவணைச் செயலி தானாகவே கூட்டுத்தொகையை திரும்பக் கணக்கிட்டு நுண்ணறை **B4–**இல் உள்ள விடையை 220 ஆக மாற்றிக் காட்டுவதைக் காணலாம். (படம் 6.3)



படம் 6.3 அட்டவணைச் செயலியில் திரும்பச் செய்த கணக்கீட்டைக் காட்டும் திரை

மின் அட்டவணைச் செயலிகள், அட்டவனையில் உள்ள தரவுகளை மிகவும் கவர்ச்சிகரமான முறையில் மனத்தில் நிற்க வைக்கும் வகையில், பட்டை வரைபடங்களாகவோ, வட்ட வரை படங்களாகவோ, கோட்டு வரைபடங்களாகவோ முப்பரிமாண படங்களாகவோ, தவிர வேறு விதமான காட்சி வடிவங்களாகவோ காட்ட வல்லவைகளாகும்.

தொடக்கத்தில் **Spread Sheet** (அட்டவணைச் செயலி), **Work Sheet** (அட்டணைத்தாள்) அல்லது அட்டவணைக் கோப்பு) என்ற இரு சொற்களும் ஒரே பொருளைத்தான் குறித்தன. ஆனால், நாளடைவில் **Spread Sheet** என்ற சொல் மின் அட்டவணையைச் செயல்படுத்தும்

மென்பொருள் தொகுப்பை **(Software Package)** மட்டுமே குறிப்பிடத் தொடங்கின. **Work Sheet** என்ற சொல் **Spread Sheet**– ஐக் கொண்டு உருவாக்கப்படும் அட்டவணையைக் கொண்ட கோப்புகளைக் குறிக்கும். எனவே நாம் **Spread Sheet** என்ற சொல் 'அட்டவணைச் செயலி' என்ற மென்பொருள் தொகுப்பைக் குறிக்கவும், **Work Sheet** என்ற சொல் அட்டவணைத் தாள் (அல்லது அட்டவணைக் கோப்பு)களைக் குறிக்கவும் பயன்படுத்துவோம்.

6.1.2 அட்டவணைச் செயலியின் பயன்பாடுகள்

அட்டவணைச் செயலியை ஏராளமான செயல்பாடுகளுக்குப் பயன்படுத்தலாம். அவற்றில் சில கீழே கொடுக்கப்பட்டுள்ளன:

- சம்பளப் பட்டியல் தயாரித்தல்
- வருமானவரிக் கணக்கீடுகள் செய்தல்
- விலைப்பட்டியல்கள் தயாரித்தல்
- கணக்கு அறிக்கைகளை உருவாக்குதல்
- பொருட்பட்டியல் தயாரித்தல்
- விலை– பயன் பகுப்பாய்வு செய்தல் **(Cost - Benefit Analysis)**
- பொருள்நிலை கணக்கீடுகள் செய்தல் **(Financial Accounting)**
- ஒப்பந்தப்புள்ளி மதிப்பீடுகள் செய்தல் **(Tender Evaluation)**
- மாணவர் தேர்வறிக்கை ஆய்வு செய்தல்

6.1.3 அட்டவணைச் செயலியால் விளையும் நன்மைகள்

மின் அட்டவணைச் செயலி, கையால் செயல்படுவதை விட பல நன்மைகளைக் கொண்டுள்ளது. சில நன்மைகள் கீழே கொடுக்கப்பட்டுள்ளன:

- அட்டவணைச் செயலியில் உருவமைக்கப்பட்டுள்ள எண் கணித, பொருளாதார, மேலும் புள்ளி விவரக் கணக்குச் சார்புகள் **(Functions)** மூலம் கணிப்பீடுகள் தானே செயல்படுத்தப்படுகின்றன.
- தேவையான அளவிற்கு துல்லியமான விடைகளைப் பெற உதவுகின்றன.
- அட்டவணைக் கோப்புகளை **(WorkSheet)**, வேண்டிய அளவிற்குப் பெரியதாக அமைத்துக்கொள்ளலாம்.
- அட்டவணைக் கோப்பின் எந்தப் பகுதியை வேண்டுமானாலும் பார்க்கவோ, பதிப்பாய்வு செய்யவோ முடியும்.
- அட்டவணைக் கோப்பின் எந்தப் பகுதியை வேண்டுமானாலும் இன்னொரு புதிய அல்லது பழைய அட்டவணைக் கோப்புடன் ஒன்று சேர்க்க முடியும்.
- அட்டவணைக் கோப்பின் எந்தப் பகுதியை வேண்டுமானாலும் விரும்பிய வடிவில் அச்சிட்டுப் பெற முடியும்.

- அட்டவணைக் கோப்பின் தரவுகளை வரைபடங்களாகவோ, வண்ணப்படங்களாகவோ பார்க்க முடியும்.
- அட்டவணைக் கோப்பின் தகவல்களை பிற தரவுத் தளங்களுக்கோ **(Database)**, சொற்செயலிகளுக்கோ **(Word Processor)** மாற்ற முடியும்.

6.1.4 வழக்கில் உள்ள அட்டவணைச் செயலிகள்

அட்டவணைச் செயலிகள் 1979–ஆம் ஆண்டு முதல் பயன்பாட்டில் உள்ளன. **VisiCalc** (விசிகால்க்) என்பது முதன் முதலில் கண்டிபிடிக்கப்பட்ட அட்டவணைச் செயலியாகும். இது 63 நெடுவரிசைகளையும் 254 வரிசைகளையும் கொண்டுள்ளது. **VisiCalc.** (விசிகால்க்) முக்கியமாக பொருளாதார கணக்காய்வுச் சார்புகளைக் கொண்ட மென்பொருளாகும்.

'விசிகால்க்' – இன் ஆற்றலை உணர்ந்தபின் பல கணிப்பொறி நிறுவனங்கள் வெவ்வேறு அட்டவணைச் செயலிகளை உருவாக்கத் தொடங்கினர். **Lotus Corporation** (லோட்டஸ் கார்ப்ரேஷன்) 1982–இல் **LOTUS 1–2–3** (லோட்டஸ் 1–2–3) என்ற அட்டவணைச் செயலியை வெளியிட்டது. இது வரைபட வசதியையும் கொண்டு இருந்ததால், மக்களிடையே மிகப்பெரும் வரவேற்பைப் பெற்றது.

LOTUS 1–2–3 (லோட்டஸ் 1–2–3)–ஐ அடுத்த **Excel** (எக்ஸல்) என்ற அட்டவணைச் செயலியை **Microsoft Corporation** (மைக்ரோ சாப்ஃட் கார்ப்ரேஷன்) என்ற நிறுவனமும், **Quattro Pro** (க்வாட்ரோப்ரோ) என்ற அட்டவணைச் செயலியை **Borland International** (போர்லாண்டு இண்டர்நேஷனல்) என்ற நிறுவனமும், **Improve** (இம்ப்ரோவ்) என்ற மென்பொருளை **LOTUS Corporation** (லோட்டஸ் கார்பரேஷன்) –னும், **StarCalc** (ஸ்டார் கால்க்) என்ற அட்டவணைச் செயலியை **Sun Micro Systems** (சன் மைக்ரோ சிஸ்டம்ஸ்) என்ற நிறுவனமும் உருவாக்கி வழங்கியுள்ளன.

6.2. ஸ்டார் ஆபீஸ் (Star office)–இன் ஸ்டார் கால்க் (StarCalc) ஐ பயன்படுத்துவது எப்படி ?

StarCalc (ஸ்டார் கால்க்) என்னும் அட்டவணைச் செயலி **Star office** என்னும் மென்பொருளைச் சேர்ந்த ஒரு பகுதியாகும். **StarCalc** –ஐ பயன்படுத்த வேண்டுமானால், முதலில் **"Star office"** என்ற மென்பொருளை ஏதாவது ஒரு Windows (விண்டோஸ்) இயக்க அமைப்பு மென்பொருளுடன், கணிப்பொறியில் உள்ளிட வேண்டும்.

StarCalc ஒரு திறன் மிக்க அட்டவணைச் செயலியாகும். இது வணிகம் மற்றும் பல பயன்பாடுகளுக்குத் தேவையான பொருள்நிலைச் சார்புகள், புள்ளிவிவரச் சார்புகள், தரவுத்தள அமைப்புச் சார்புகள் போன்ற எல்லா சார்புகளையும் **(Functions)** கொண்டுள்ளது.

அட்டவணைக் கோப்பை **(WorkSheet)** உருவமைக்க வேண்டுமானால், தரவுகளை ஒழுங்குபடுத்துதல், தரவுகளை உள்ளிடல், வாய்பாடுகளை உண்டு பண்ணுதல், அட்டவணைக் கோப்பை பதிப்பாய்வு செய்ய, எண்கள், அடையாளங்கள், நுண்ணறைகள் போன்றவற்றை வடிவமைத்தல் போன்ற பல பணிகளை படிப்படியாக செய்தல் வேண்டும். தேவையெனில், வரைபடங்களை இணைத்தல், தரவுகளை ஆய்வு செய்தல், அட்டவணைக் கோப்புகளை அச்சிடல் போன்ற பணிகளையும் செய்ய வேண்டியிருக்கும்.

இப்பொழுது **StarCalc** –ஐ பயன்படுத்தி அட்டவணைக் கோப்புகளை எப்படி உண்டு பண்ணுவது என்று பார்ப்போம்.

6.2.1 முதல் அட்டவணைத் தாளை உண்டு பண்ணுவது எப்படி ?

StarCalc–இல் வேலை செய்ய **Star Office** மென்பொருளை இயக்கி உள்ளே சென்று முதன்மைப் பணிப் பட்டியல் பட்டை **(Main Menu Bar)** யில் உள்ள **File** என்ற பணிப்பட்டியை **(Menu)** அழுத்தவும். பிறகு**'New'** என்ற பட்டியையும் அடுத்து **Spreadsheet** என்ற பட்டியையும் அழுத்தவும். உடனே படம் **6.1** இல் காட்டியவாறு புது அட்டவணைத்தாள் திறப்பதைக் காணலாம்.

சன்னல் திரையின் **(Windows)** மேற்பகுதியில் பணிப்பட்டியல் பட்டை **(Menu Bar)** உள்ளது. பணிப்பட்டியலில் உள்ள ஒரு பட்டியை **(Menu)** பயன்படுத்த சுட்டெலியைக் **(Mouse)** கொண்டு சுட்டியை **(Cursor)** அந்த பட்டியின் மேல் எடுத்துச் சென்று, சுட்டெலியின் இடப்புற பொத்தானை அழுத்த வேண்டும். உடனே பணிக்குறிகள் **(icons)** பலவற்றைக் கொண்ட பணிகுறிப்பட்டியல் தோன்றும். அதிலிருந்து நாம் எந்த பணியை செயல்படுத்த விரும்புகின்றோமோ அந்த பணிக்குறியின் **(Icon)** மேல் சுட்டியை வைத்து சுட்டெலியைத் தட்டினால், அந்த பணி செயல்படத் தொடங்கும் பணிப்பட்டியல் பட்டைக்குக் கீழே சார்புக்குறிப்பட்டை **(Functional Bar)**யும் பொருள் பட்டையும் **(Object Bar)** உள்ளன. இந்தப் பட்டைகளில் அடிக்கடி பயன்படும் செயற்கூறுகளை விரைவில் செய்ய வல்ல பணிக்குறிகள் **(Icons)** உள்ளன. இந்தப் பட்டைகளுக்குக் கீழே வாய்பாட்டுப்பட்டை **(Formula Bar)** உள்ளது. இந்தப் பட்டைகளுக்குக் செயற்பாட்டிலுள்ள நுண்ணறையில் உள்ளத் தரவுகளை வெளிக்காட்ட பயன்படுகிறது. இத்துடன், மேலும் சில பணிக்குறிகளும் உள்ளன. வாய்பாட்டுப் பட்டைக்குக் கீழே, அட்டவணைக் கோப்பின் நெடுவரிசைத் தலைப்புகள் **(Column Headings)** உள்ளன. அடிப்புறத்தில் முதன்மைக் கருவிப்பட்டை **(Main Tool Bar)** உள்ளது. அதனையடுத்து வரிசைத் தலைப்புகள் **(Row Headings)** உள்ளன. கடைகோடியில் நிலைப்பட்டை **(Status Bar)** உருள்பட்டைகளும் **(Scroll Bar)** உள்ளன.

முதல் அட்டவணைத் தாளை **(Work Sheet)** உருவாக்கத் தொடங்குமுன், தேவையான தரவுகளை சேகரிக்க வேண்டும். எடுத்துக்காட்டாக, மாணவர்களின் மதிப்பெண்களைக் கொண்ட ஒரு அட்டவணைக் கோப்பை உருவாக்க விரும்பினால், மாணவர்களின் பதிவெண், பெயர், பாடப்பெயர்கள், ஒவ்வொரு பாடத்திலும் மாணவர்கள் ஒவ்வொருவரும் பெற்ற மதிப்பெண்கள் போன்ற தரவுகளைச் சேகரிக்க வேண்டும். பிறகு இந்தத் தரவுகளை அட்டவணைத்தாளில் எழுத வேண்டும். அதாவது அட்டவணைக் கோப்பினுள் இட வேண்டும். தரவுகளை அட்டவணைத்தாளில் எழுதிய பின், இத்தரவுகளைக் கொண்டு மொத்த மதிபெண்கள் கணக்கிடுதல் அல்லது சராசரி மதிப்பெண்களைக் கணக்கிடுதல் போன்ற கணக்கீடுகளைச் செய்யலாம். இந்தத் தரவுகள் அட்டவணைத் தாளில் காட்சி அளிக்குமாறு இவற்றை வடிவமைத்துக் கொள்ளலாம். மேலும் இத்தரவுகளை வரைபட வடிவிலும் வரைந்து காட்டலாம்.

6.2.2 அட்டவணைக் கோப்பில் தரவுகளைக் கொடுப்பது எப்படி ?

தரவுகளைச் சேகரித்த பின் அவற்றை அட்டவணைக் கோப்பினுள் இட தயாராகி விட்டீர்களா ? தரவைப் பெற அணியமாக உள்ள நுண்ணறையை அறிய நுண்ணறைச்சுட்டி **(Cell Pointer)** உதவுகிறது. நுண்ணறைச்சுட்டி என்பது நுண்ணறையைச் சுற்றி பளிச்சென்று தெரியும் ஒரு செவ்வக வளையம் ஆகும். அட்டவணைக் கோப்பில் பணிசெய்யத் தொடங்கும்போது எப்பொழுதுமே தொடக்கத்தில் நுண்ணறைச் சுட்டி **A1** இல் தான் இருக்கும். நுண்ணறைச்சுட்டியின் இடத்தை , சுட்டெலியைக் கொண்டோ அல்லது விசைப் பலகையில் உள்ள அம்புப் பொத்தான்களைக் **(arrow keys)** கொண்டோ மாற்றலாம்.

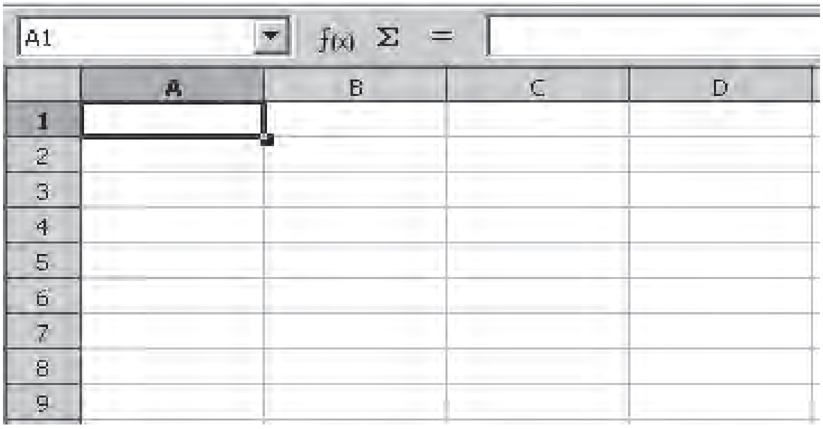
விசைப்பலகையில் உள்ள **Tab,Home, End, PgUp, PgDown** போன்ற பொத்தான்களைக் கொண்டு நுண்ணறைச் சுட்டியை அட்டவணைத்தாளின் எந்த இடத்திற்கு வேண்டுமானாலும் எடுத்துச் செல்லலாம். நுண்ணறையைத் தேர்ந்தெடுத்துக் கொண்ட பின் தரவுகளைக் கொடுக்கலாம். கொடுக்கப்படும் தரவு எண்ணாக இருந்தால், அட்டவணைச் செயலி அதனை தானாக எண் என்று அறிந்துகொண்டு அதன் மீது கணக்கீடுகள் செய்ய அனுமதிக்கின்றது. கொடுக்கப்படும் தரவு 'சொல்'லாக இருந்தால் அதனை ஒரு அடையாளமாக (Label) அறிந்துகொண்டு அதன் மீது கணக்கீடுகள் செய்ய அனுமதிக்காது.

StarCalc அட்டவணைச் செயலி, அட்டவணைக்கோப்பில் தேதி, நேரம் போன்றவற்றையும் உள்ளிட வழி செய்கின்றது. இது மிகவும் பயனுள்ளதாகும். இதனைக் கொண்டு, ஆண்டு, மாதம், நாள், நேரம் தொடர்பான கணக்கீடுகளைச் செய்துகொள்ளலாம். எடுத்துக்காட்டாக ஒருவருடைய வயதைக் கணக்கிட இரண்டு தேதிகளை பயன்படுத்தி ஒன்றிலிருந்து ஒன்றை கழிக்க வேண்டும்; ஒவ்வொரு வாரமும் வரும் திங்கட்கிழமை என்ன தேதிகளில் வருகின்றன என்று அறிய ஒரு தேதியுடன் ஏழு நாட்களைக் கூட்ட வேண்டும். நுண்ணறைக் கோப்பில் எண்ணையும், சொற்களையும் கொடுத்ததைப் போலவே, தேதிகளையும் கொடுக்கலாம். இதனைச் செய்ய சுட்டியை குறிப்பிட்ட நுண்ணறைமேல் வைத்துத் தேர்ந்து, பிறகு தேதியை மாதம்/ நாள்/ ஆண்டு **(MM/DD/YY)** என்ற வரிசைப்படி கொடுக்க வேண்டும். இது பற்றி மேலும் விரிவாக பின்வரும் பகுதிகளில் படிப்போம்.

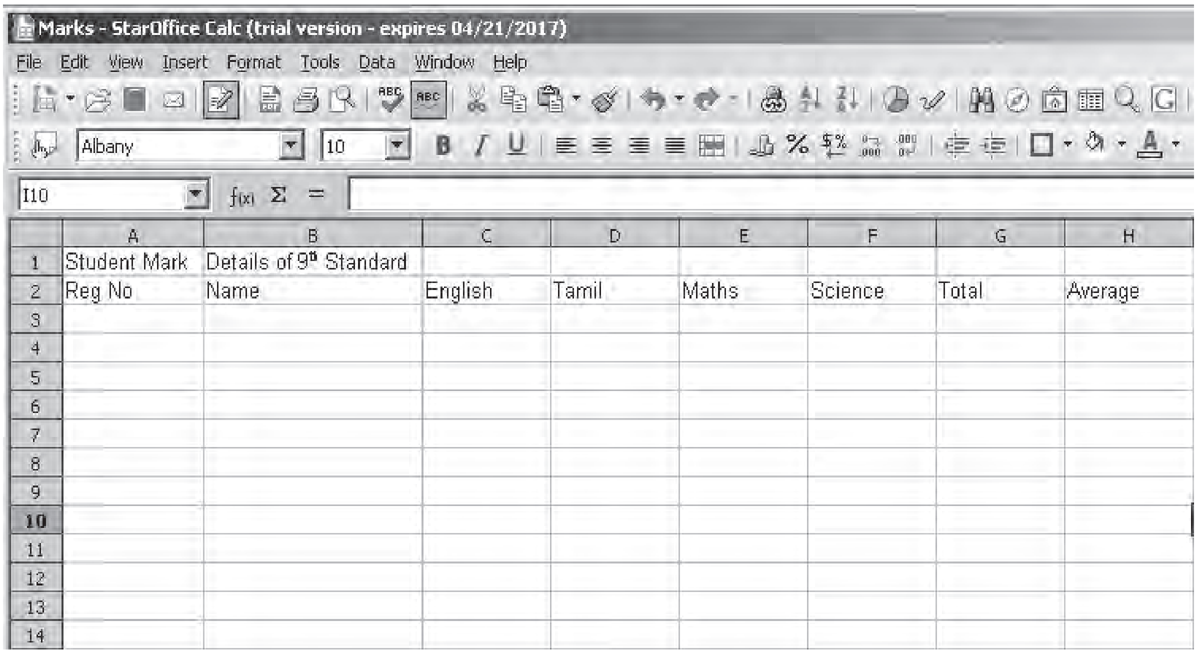
மாணவர்களின் மதிப்பெண் பட்டியலை அட்டவணைக் கோப்பில் எப்படி செயல்படுத்துவது என்பதற்கான வழிமுறைகள் கீழே கொடுக்கப்பட்டுள்ளன. தொடக்கத்தில் சுட்டி **A1** இல் இருக்கும் என்பதை நினைவில் கொள்க. (படம் 6.4)

- நுண்ணறை **A1**–இல் **'Student Mark Details of 9th Standard'** என்ற அட்டவணையின் தலைப்பை அச்சிட்டு உள்ளிடவும். பிறகு கீழ்நோக்கு அம்புப் பொத்தானை **(Down Arrow Key)** அழுத்தி நுண்ணறை **A2**–க்குச் செல்லவும்.
- A2 –இல் **'Reg.No'** என்ற தலைப்பைக் கொடுக்கவும். பிறகு வலம் நோக்கு அம்புப் பொத்தானை அழுத்தி நுண்ணறை **B2**–க்குச் செல்லவும்.
- **B2** –இல் **'Name'** என்று கொடுக்கவும் பிறகு C2–க்குச் செல்லவும்.
- **C2** –இல் பாடப் பெயராகிய **'English'** என்ற சொல்லைக்கொடுக்கவும்.
- **D2** –இல் **'Tamil'** என்று கொடுக்கவும்.
- **E2** –இல் **'Maths'** என்று கொடுக்கவும்.
- **F2** –இல் **'Science'** என்று கொடுக்கவும்.
- **G2** –இல் **'Total'** என்று கொடுக்கவும்.
- **H2** –இல் **'Average'** என்று கொடுக்கவும்.

இப்பொழுது அட்டவணைக் கோப்பு உருவாகிவிட்டது. இப்படி உருவாக்கப்பட்ட அட்டவணைத்தாள் படம் 6.5 –இல் காட்டப்பட்டுள்ளது.



படம் 6.4 காலி அட்டவணைத் தாள்



படம் 6.5 மாணவர் மதிப்பெண் பட்டியலுக்கான அட்டவணை

6.2.3 அட்டவணைக் கோப்பை சேமித்தல்

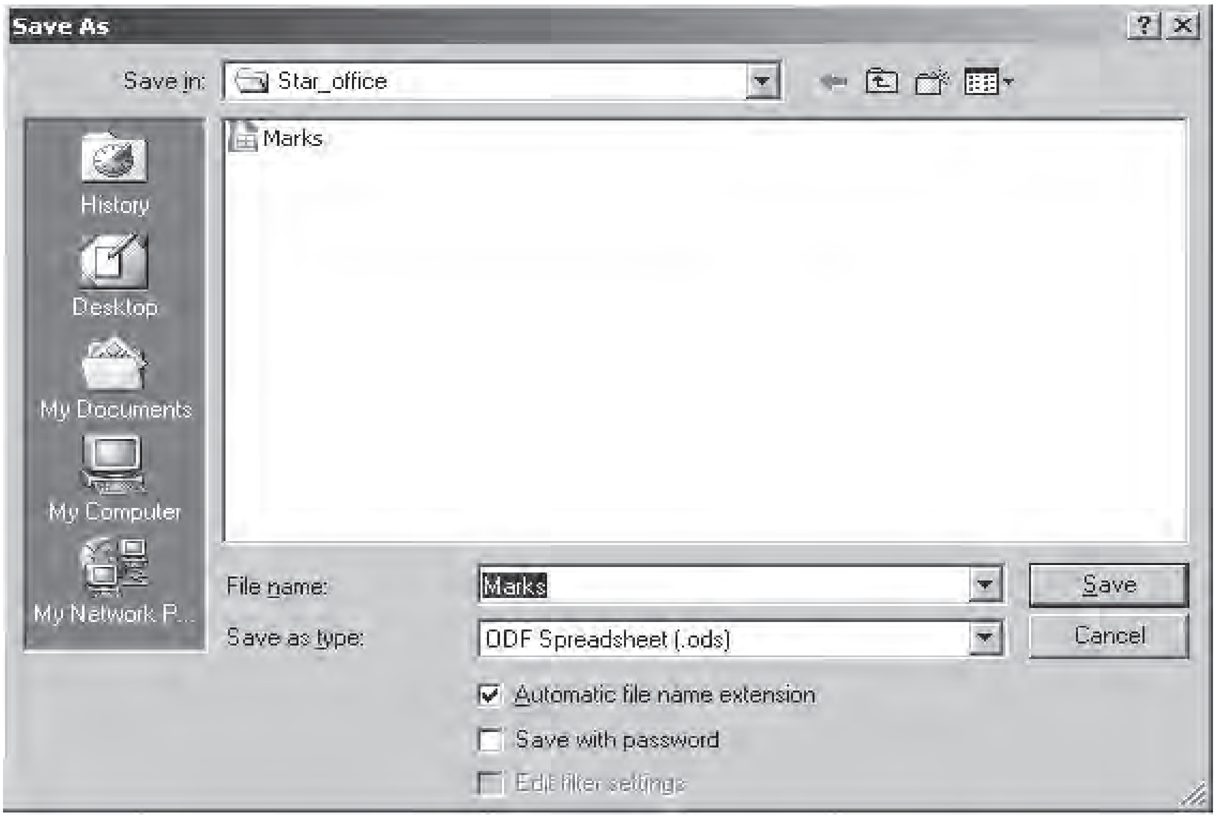
மேற்கண்டவாறு உருவாக்கிய அட்டவணைக் கோப்பை சேமிக்க, **'File'** என்ற பணிப்பட்டிக்குள் சென்று **'Save'** அல்லது **'Save As'** என்ற பட்டியை தேர்வு செய்யவும். படம் **6.6** இல் காட்டியுள்ளது போன்ற ஒரு திரை தோன்றும். இப்பொழுது, அட்டவணைக் கோப்பை எந்த பெயரில் சேமிக்க விரும்புகிறீர்களோ அந்த பெயரைத் திரையில் சுட்டியுள்ள இடத்தில் அச்சிட்டு **'Save'** என்ற பணிப்பட்டியை சுட்டெலியைக் கொண்டு 'கிளிக்' செய்து சேமிக்க. எடுத்துக்காட்டாக, மேலே உருவமைத்த மாணவர் மதிப்பெண் பட்டியலை **'Marks'** என்ற பெயரில் சேமிக்க, **'Marks'** என்ற சொல்லை அச்சிட்டு உள்ளிட்டும். **'Save'** பொத்தானை அழுத்தி கோப்பைச் சேமிக்க வேண்டும். சார்ப்புப் பட்டையில் **(Functional Bar)** உள்ள இந்த பணிக்குறியை 'கிளிக்' செய்தும், சேமிக்கலாம்.

6.2.4 அட்டவணைக் கோப்பை முடித்தல்

அட்டவணைக் கோப்பின் செயல்பாட்டை முடித்துக் கொண்டு வெளியே வர **‘File’** என்ற பணிப்பட்டிக்குச் சென்று **‘Close’** என்ற பட்டியைத் தேர்ந்து சுட்டெலியையோ அல்லது விசைப்பலகையின் **‘Enter’** பொத்தானையோ அழுத்த வேண்டும்.

6.2.5 அட்டவணைக் கோப்பைத் திறத்தல்

ஏற்கனவே சேமிக்கப்பட்டுள்ள அட்டவணைக் கோப்புகளை மீண்டும் திறக்க.**‘File’** என்னும் பணிப்பட்டிக்குள் சென்று **open** என்ற பட்டியைத் தேர்வு செய்யவும். உடனே கோப்புகளின் பெயர்ப் பட்டியலைக் கொண்ட உரையாடல் பெட்டி **(Dialog Box)** ஒன்று தோன்றும். அதில் தேவையான கோப்பை தெரிவு செய்து **open** என்ற பணிப்பட்டியை ‘கிளிக்’ செய்யவும். மாறாக சார்புப்பட்டையில் உள்ள இந்த பணிக்குறியை கிளிக் செய்வதன் மூலம் ஏற்கனவே சேமிக்கப்பட்டுள்ள கோப்பைத் திறக்கலாம்.



படம் 6.6 save As உரையாடல் பெட்டி

6.2.6 Star Office – லிருந்து வெளியேறல்

‘File’ பட்டிக்குள் உள்ள **Exit** என்ற பட்டியை **(Menu)** பயன்படுத்தி **Star Office–** லிருந்து வெளியேறலாம்.

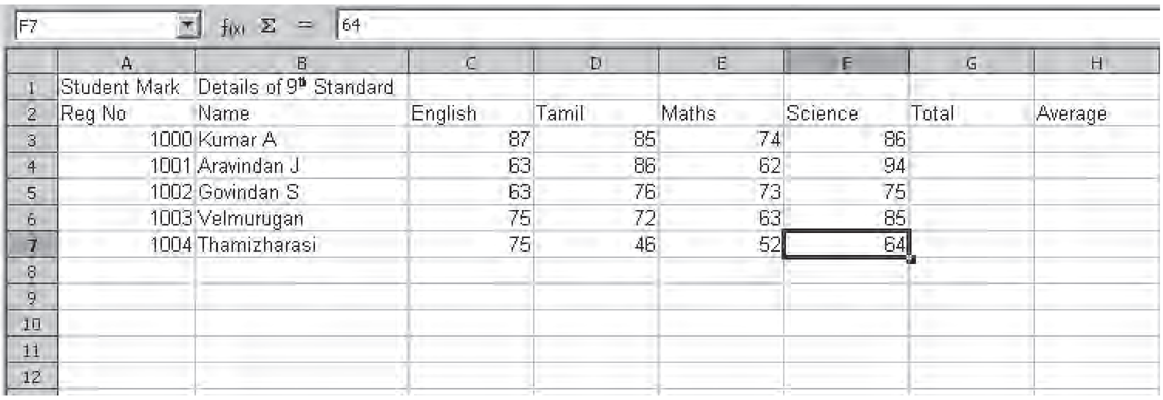
செய்து பார்

கீழ்க்கண்ட மதிப்பெண்களுக்கான அட்டவணைக் கோப்பை மேற்கண்ட பத்திகளில் கூறியுள்ள முறைகளைப் பின்பற்றி உருவாக்கி அதனை **Marks** என்ற பெயரில் சேமிக்கவும்.

Reg No.	Name	English	Tamil	Maths	Science
1000	Kumar A.	87	85	74	86
1001	Aravindan J.	63	86	62	94
1002	Govindan S.	63	76	73	75
1003	Velmurugan T.	75	72	63	85
1004	Thamizharasi G.	75	46	52	64

நீங்கள் உருவாக்கிய அட்டவணைத்தாள் படம் **6.7** –இல் காட்டியுள்ளவாறு உள்ளதா ? செய்து பாருங்கள்.

குறிப்பு : மேற்கூறியவாறு உருவாக்கிய அட்டவணைத் தாளில் **B6&B7** நுண்ணறைகளில் முழுப் பெயர் தெரியவில்லை பாருங்கள். இது ஏன் தெரியுமா ? இதற்குக் காரணம் **B** நெடுவரிசையின் நுண்ணறைகளின் அகலம் போதிய அளவு இல்லை. இதனை **StarCalc** ஒரு சிறு சிவப்பு முக்கோணத்தின் மூலம் வெளிப்படுத்துகின்றது. இதனை எப்படி பெரிதுபடுத்துவது என்பதை இந்த அத்தியாயத்தின் பின் பகுதிகளில் படித்து அறிந்துகொள்வீர்கள்.


F7 | 64

	A	B	C	D	E	F	G	H
1	Student Mark	Details of 9th Standard						
2	Reg No	Name	English	Tamil	Maths	Science	Total	Average
3	1000	Kumar A	87	85	74	86		
4	1001	Aravindan J	63	86	62	94		
5	1002	Govindan S	63	76	73	75		
6	1003	Velmurugan	75	72	63	85		
7	1004	Thamizharasi	75	46	52	64		
8								
9								
10								
11								
12								

படம் 6.7 மாணவர் தளத்தின் அட்டவணைக் கோப்பு

6.3 அட்டவணைக் கோப்பின் தரவுகளை பதிப்பாய்வு செய்தல்

அட்டவணைக் கோப்பில் உள்ளத் தரவுகளை நாம் விரும்பியவாறு திருத்தலாம்; நீக்கலாம், சேர்க்கலாம். வாய்பாடு பட்டையில் உள்ள வாய்பாடுகளைக் கூட திருத்தலாம். இப்படிப்பட்ட செயல்களுக்கு பதிப்பாய்வு செய்தல் **(Editing)** என்று பெயர். இவற்றைச் செய்ய **'File'** என்ற பட்டிக்குள் சென்று **'Open'** என்ற பட்டியை 'கிளிக்' செய்ய வேண்டும். பிறகு சுட்டியை **(Cursor),** நாம் எந்த நுண்ணறையில் உள்ள தரவை (எண் அல்லது சொல்) மாற்ற விரும்புகின்றோமோ அந்த நுண்ணறைக்குச் சென்று கிளிக் செய்ய வேண்டும்.

தரவுகளை பதிப்பாய்வு செய்ய கீழ்க்காணும் இருமுறைகளைப் பயன்படுத்தலாம்.

1. சுட்டியை தரவு மாற்றப்பட வேண்டிய நுண்ணறைக்கு எடுத்துச் சென்று கிளிக் செய்க. பிறகு புது தரவை அச்சிடுக. புதுத்தரவு பழைய தரவின் மீது எழுதப்படுவதைக் காணலாம்.

2. வாய்பாட்டைத் திருத்த விரும்பினால், வாய்பாட்டுப் பட்டையில் 'கிளிக்' செய்யவும். ஒரு நெடுக்கோட்டுச் சுட்டி **(Vertical Cursor)** வாய்பாட்டின் மேல் தோன்றும். இப்பொழுது,

விசைப்பலகையில் உள்ள இடநோக்கு அம்புப் பொத்தான் அல்லது பிற்போக்குப் பொத்தானை **(Backspace)** அழுத்தி சுட்டியை நகர்த்தி வாய்பாட்டை புதுப்பிக்கலாம்.

6.4 வாய்பாடுகளை அட்டவணையில் உள்ளிடல்

அட்டவணைக் கோப்பில் தரவுகளைக் கொடுத்த பின் அவற்றின் மீது கணக்கீடுகளைச் செய்யலாம். இதற்கு வாய்பாடுகளை அட்டவணையில் உள்ளிட வேண்டும்.வாய்பாடுகளை அட்டவணையில் உள்ளிட **StarCalc -** இல் உருவாக்கப்பட்டுள்ள, இது தொடர்பான இலக்கண அமைப்பு பற்றி தெரிய வேண்டும். **StarCalc -** ல் உள்ள இலக்கணப்படி, வாய்பாடுகள் '=' என்ற சமக்குறியுடன் தொடங்க வேண்டும், பிறகு சமக்குறியைத் தொடர்ந்து எண்கள் கணக்கீட்டுக் குறிகள், நுண்ணறை முகவரி போன்றவற்றைக் கொடுக்க வேண்டும். எடுத்துக்காட்டாக, **A5 –**இல் உள்ள எண்ணை **4–**ல் பெருக்கி, அத்துடன் **B5 –**இல் உள்ள எண்ணைக் கூட்டி வரும் விடையை **C5–**இல் இட வேண்டும் என்று கொள்வோம். இதற்கான வாய்பாட்டைக் கீழ்க்கண்டவாறு **C5–**இல் சுட்டியை கொண்டு சென்று கொடுக்க வேண்டும்.

= (4* A5 + B5)

StarCalc-இல் உள்ள பல வகையான இயக்கிகள் **(Operators)** கீழே கொடுக்கப்பட்டுள்ளன.

6.4.1 எண் கணித இயக்கிகள் (Arithmetic Operators)

கீழ்க்கண்ட இயக்கிகள் விடைகளை எண்களாக கொடுக்கும்.

இயக்கிகள்	**பெயர்**	**எடுத்துக்காட்டு**
+(plus)	Addition	1+1
-(minus)	Subtraction	2-1
-(Minus)	Negation	-5
*(asterisk)	Multiplication	2*2
/(Slash)	Division	9/3
% (Percent)	Percent	15%
∧ (Caret)	Exponentiation	3∧2

6.4.2 ஒப்பீடு (Comparative) இயக்கிகள்

இந்த இயக்கிகள் உண்மை **(True)** அல்லது பொய் **(False)** என்ற தகவலைக் கொடுக்கும்.

இயக்கிகள்	**பெயர்**	**எடுத்துக்காட்டு**
=	Equal	A1=B1
>	Greater than	A1>B1
<	Less than	A1<B1
> =	Greater than or equal to	A1>=B1
< =	Less than or equal to	A1<=B1
< >	Inequality	A1<>B1

6.4.3 உரை (Text) இயக்கிகள்

இந்த இயக்கி உரைப்பகுதிகளை ஒன்றாக இணைக்க உதவுகின்றது.

இயக்கிகள்	பெயர்	எடுத்துக்காட்டு
& (AND)	உரை இயக்கி **(Text Operator)**	**"Star" & " Office"** என்று கொடுத்தால் **"Star Office"** என்று வரும்.

6.4.4 பார்வையிடு (Reference) இயக்கிகள்

இந்த இயக்கிகள் ஒரு முகவரியிலிருந்து இன்னொரு முகவரி வரைப் பரவியுள்ள நுண்ணறைகள் அனைத்தையும் கணிப்பீட்டிற்கு உள்ளாக்குகின்றன.

இயக்கிகள்	பெயர்	எடுத்துக்காட்டு
: (Colon)	பரப்பு **(Range)**	**A1:C108** **(A1**முதல்**C108** வரை உள்ள நுண்ணறைகளைக் குறிக்கும்.)
! (Exclamation)	குறுக்கீடு **(Intersection)**	**SUM (A1:B6 ! B5: C12)**

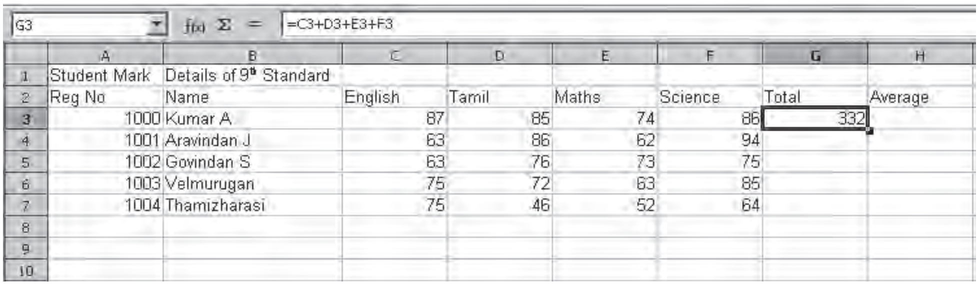
எண் கணித இயக்கிகளை வாய்பாடுகளில் பயன்படுத்தும் போது கணக்கியலில் பயன்படுத்தப்படும் அதே வரிசை முறையில்தான் **StarCalc**–லும் கணிப்பீடுகள் செய்யப்படுகின்றன. கணிப்பீட்டு வரிசை முறை கீழ்க்கண்டவாறு இருக்கும்:

1. திறன் **^ (Exponentiation)**
2. எதிர் நிலையாக்கல் **(Negative)**
3. பெருக்கல் **&** வகுத்தல் **(*, /)**
4. கூட்டல் **&** கழித்தல் **(+. –)**

வாய்பாட்டை அட்டவணைக் கோப்பில் எப்படி உருவாக்குவது என்பதை கீழ்க்கண்ட எடுத்துக்காட்டு மூலம் தெளிவாக அறியலாம்.

- எந்த நுண்ணறையில் வாய்பாட்டை உள்ளிட வேண்டுமோ அந்த நுண்ணறை மீது நுண்ணறைச்சுட்டியை எடுத்துச் செல்லவும். மாணவர் மதிப்பெண் அட்டவணைத் தாளில் நுண்ணறை **G3–**இல் மொத்த மதிப்பெண்ணைக் கூட்டி எழுத வேண்டுமல்லவா ? எனவே **G3–**இல் நுண்ணறைச் சுட்டியை வைக்கவும்.
- பிறகு **= C3 + D3 + E3 + F3** என்ற வாய்பாட்டை அச்சிட்டு **'Enter'** பொத்தானை அழுத்தவும். உடனே **Kumar A** என்ற மாணவரின் மொத்த மதிப்பெண் **G3** என்ற நுண்ணறையில் தோன்றுவதைக் காணலாம்.

இப்படிக் கணக்கிட்டு உருவாக்கிய மதிப்பெண் அட்டவணைத்தாள், படம் **6.8** இல் காட்டப்பட்டுள்ளது.



G3 =C3+D3+E3+F3

	A	B	C	D	E	F	G	H
1	Student Mark	Details of 9th Standard						
2	Reg No	Name	English	Tamil	Maths	Science	Total	Average
3	1000	Kumar A	87	85	74	86	332	
4	1001	Aravindan J	63	86	62	94		
5	1002	Govindan S	63	76	73	75		
6	1003	Velmurugan	75	72	63	85		
7	1004	Thamizharasi	75	46	52	64		
8								
9								
10								

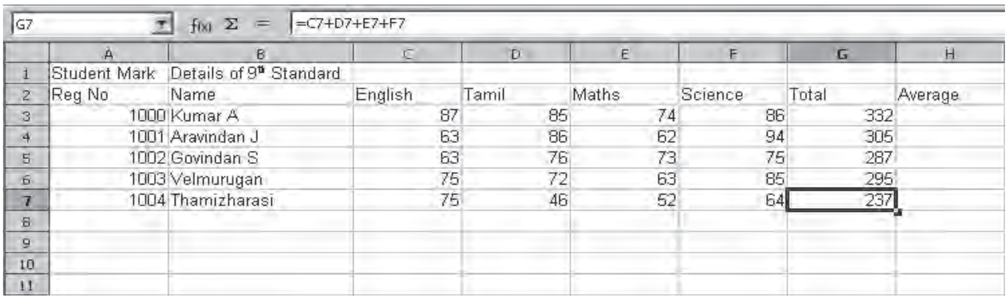
படம் 6.8 வாய்பாடு மூலம் கணக்கிட்ட மொத்த மதிப்பெண்ணுடன் உள்ள மாணவர் தரவுத்தளம்.

6.5 Fill கட்டளை

இதுவரை கணக்கீடுகளைச் செய்ய வாய்பாடுகளை எப்படி உருவாக்குவது, அவற்றை எப்படி பயன்படுத்துவது என்பது பற்றி அறிந்தோம். மேற்கண்ட எடுத்துக்காட்டில், ஒரே ஒரு மாணவனின் மொத்த மதிப்பெண்ணை மட்டும் கணக்கிட்டோம். மற்ற மாணவர்களின் மொத்த மதிப்பெண்களையும் மேற்கண்ட முறையையே பயன்படுத்தி **G4,G5,G6&G7** நுண்ணறைகளில் வாய்பாட்டை எழுதி கணக்கிட்டு எழுதலாம். ஆனால் இதைவிட ஒரு எளிய முறை உள்ளது. வாய்பாட்டை நுண்ணறை **G3** யில் எழுதி, பிறகு அதனை பிற நுண்ணறைகளில் நகலாக்கலாம். இதனை சார்புக் கருவிக்குறிப் பட்டை **(Functional Tool Bar)** யில் உள்ள **Copy & Paste** பணிக்குறிகளைக் கொண்டும் செய்யலாம். இது பற்றி Star Writer அதிகாரத்தில் படித்ததை நினைவு கூர்க. இந்த செயற்பாட்டை, **Star Calc–**இல் உள்ள **Auto Fill** என்ற கட்டளையைக் கொண்டும் செய்யலாம்.

Star Calc –இல் உள்ள **Auto Fill** என்ற வசதியைப் பயன்படுத்தி ஒரு நுண்ணறையில் உள்ளதை பல நுண்ணறைகளில் நகலெடுத்து ஒரு வரிசையை உருவாக்கலாம். ஒரு நுண்ணறைகள் உள்ளதை பிற நுண்ணறைகளில் நகலெடுக்க சுட்டியை அந்த நுண்ணறை மீது வைத்து 'கிளிக்' செய்து சுட்டெலியை இழுத்து நகலெடுக்க வேண்டிய நுண்ணறைகளை மிசைப்படுத்தவும். பிறகு **Fill → Down (or Left)** என்ற பட்டியை **Edit** பட்டிக்குள் சென்று தேர்வு செய்யவும். உடனே முதல் நுண்ணறயில் உள்ள பொருள் மிசைப்படுத்தப்பட்ட அத்தனை நுண்ணறைகளிலும் நிரப்பப்படும்.

எடுத்துக்காட்டாக, **G3** யில் உள்ள வாய்பாட்டை **G4, G5, G6, & G7** நுண்ணறைகளில் நிரப்ப, நுண்ணறைகள் **G3** முதல் **G7** வரை மிசைப்படுத்தவும். பிறகு **Edit → Fill → Down** என்ற பட்டிகளை 'கிளிக்' செய்யவும். **G3** யில் உள்ள வாய்பாடு பிற நுண்ணறைகளில் நகலிடப்பட்டு மாணவர்களின் ஒவ்வொருவர் மொத்த மதிப்பெண்களும் கணக்கிடப்பட்டு படம் **6.9** இல் காட்டப்பட்டுள்ளதைப் போல் செய்யப்படும்.



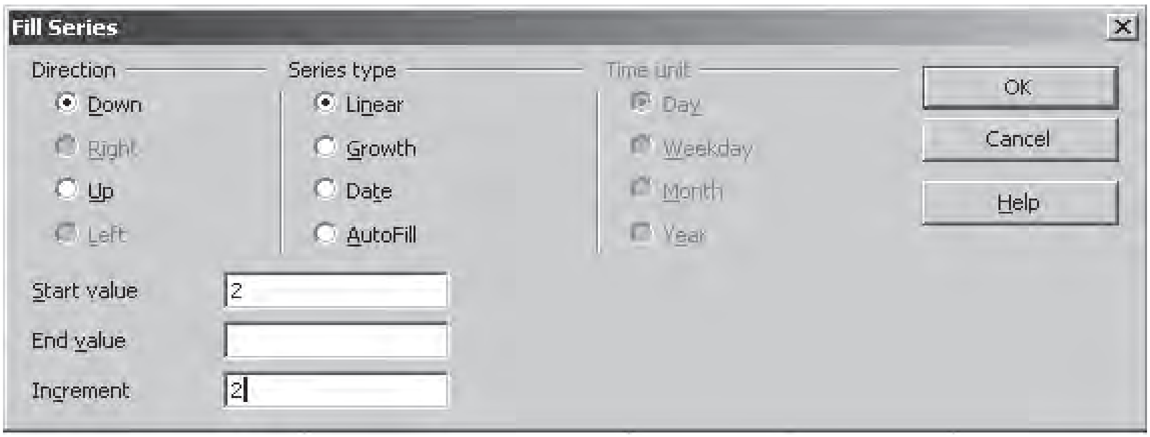
G7 =C7+D7+E7+F7

	A	B	C	D	E	F	G	H
1	Student Mark	Details of 9th Standard						
2	Reg No	Name	English	Tamil	Maths	Science	Total	Average
3	1000	Kumar A	87	85	74	86	332	
4	1001	Aravindan J	63	86	62	94	305	
5	1002	Govindan S	63	76	73	75	287	
6	1003	Velmurugan	75	72	63	85	295	
7	1004	Thamizharasi	75	46	52	64	237	
8								
9								
10								
11								

படம் 6.9 நுண்ணறை G3 யின் வாய்பாடு G4 முதல் G7 வரை நகலிடல் மூலம் கணக்கிட்டுப் பெற்ற மதிப்பெண் அட்டவணை.

குறிப்பு : அட்டவணைக் கோப்பில், ஒரு நுண்ணறையிலிருந்து இன்னொரு நுண்ணறை வரை தொடர்ச்சியாக உள்ள நுண்ணறைகளை பரப்பு **(Range)** என்று அழைப்பர். ஒரு பரப்பை **(Range)** முதல் நுண்ணறையின் முகவரி, அதனை அடுத்து ஒரு முக்கால் புள்ளி **(colon)** பிறகு இறுதி நுண்ணறையின் முகவரி ஆகியவற்றால் குறிக்கப்படுகின்றது. எடுத்துக்காட்டாக, நுண்ணறைகள் **G1, G2, G3,G4 & G5** என்ற நுண்ணறைகளின் பரப்பு **G1 : G5** என்று எழுதி குறிக்கப்படும். அதேபோல் நுண்ணறைகள் **A1,B1,C1,D1,E1 & F1** என்பது **A1: F1** என்றும் **A4,A5,A6,B4,B5 & B6** என்பது **A4 : B6** என்றும் எழுதி குறிக்கப்படும்.

Fill கட்டளையைக் கொண்டு தேர்ந்து எடுத்த நுண்ணறைகளில் தரவு வரிசைகளையும் **(Series of Data)** உருவாக்கலாம். முதலில் நிரப்ப விரும்பும் நுண்ணறைகளைத் தேர்வு செய்யவும். பிறகு **Edit → Fill → Series** பட்டிகளை 'கிளிக்' செய்யவும். இப்பொழுது திரையில் வரிசையின் வகையைத் தெரிவு செய்யவும். (படம் 6.10)



படம் 6.10 Fill Series உரையாடல் பெட்டி

எடுத்துக்காட்டாக, மதிப்பெண்களை ஏறுவரிசையில், அடுத்து அடுத்து வரும் எண்களின் வேறுபாடு 3–ஆக இருக்குமாறு, நெடுவரிசையில் உள்ள நுண்ணறைகளில் நிரப்ப விரும்புவதாகக் கொள்வோம். இதனைச் செய்ய நாம் முதல் நுண்ணறையில் மட்டும் தொடக்க எண்ணைக் கொடுத்தால் போதும், பிறகு எண்கள் மேலிருந்து கீழ்நோக்கி ஏறு வரிசையில் அமைய, **Direction** என்றப் பட்டியலில் **Down** என்ற பட்டியையும் என்ற **Type**பட்டியலில் **Increment** என்ற பட்டியையும் 'கிளிக்' செய்து **Increment** மதிப்பை 3 என்றும் கொடுத்து **OK** என்ற பட்டியை 'கிளிக்' செய். உடனே அடுத்துள்ள அத்தனை நுண்ணறைகளிலும் எண்கள் நாம் விரும்பியவாறு தானே உருவாகித் தெரிவதைக் காணலாம். (படம்6.10). இன்னொரு எடுத்துக்காட்டாக **A1:D6** என்ற **A1** முதல் **D6** வரை உள்ள நுண்ணறைகளைத் தேர்ந்து எடுக்கவும். பிறகு **Edit → Fill → Series** என்ற பட்டிகளை 'கிளிக்' செய்யவும்.

தொடக்க எண் **2** என்று **Start Value** பெட்டியிலும், மிகுக்கும் எண் **Increment Value 2** என்று **Increment** பெட்டியிலும், **Type** என்ற பெட்டியில் **Growth** என்றும், Direction என்ற பெட்டியில் Down என்றும் கொடுக்கவும். பிறகு **OK** என்ற பட்டியை 'கிளிக்' செய்யவும். உடனே படம் **6.11**–இல் காட்டிவாறு அறைகள் **A1** முதல் **D6** வரை 2^K, **K** = 0,1,2,...... என்ற வரிசை உருவாக்கப்பட்டு நிரப்பப்பட்டுள்ளதைக் காணலாம். உரையாடல் பெட்டி (Dialog Box)–இல் உள்ளதைப் போல, நீங்கள் தேதி, நேரம் இவற்றையும் தானாகவே உருவாக்கி அட்டவணையில் நிரப்பலாம். எடுத்துக்காட்டாக, **2005–** ஆம் ஆண்டு மார்ச்சு மாதம் முதல் மே மாதம் வரை உள்ள

நாட்களில் வருகின்ற ஞாயிற்றுக்கிழமைகளை பட்டியலிட்டுக் காட்டுவதாகக் கொள்வோம். இதனைச் செய்ய கீழ்க்கண்ட செய்முறையைப் பின்பற்றவும்:

- இந்த காலகட்டத்தில் முதல் ஞாயிற்றுக்கிழமையின் தேதியை **3/6/05** என்று முதல் நுண்ணறையில் கொடுக்கவும்.
- வேண்டிய அளவு நுண்ணறைகளைத் தேர்ந்து எடுத்துக் கொள்ளவும். (இந்த எடுத்துக்காட்டிற்கு 13 இடங்கள் தேவை)
- **Edit → Fill → Series** என்ற பட்டிகளைத் தேர்ந்து 'கிளிக்' செய்யவும்.
- உரையாடல் பெட்டி **(Dialog Box)** யில் **Date** அலகிற்கு Day என்றும் **– increment –** ஐ 7 என்றும் கொடுத்து **OK** பட்டியை 'கிளிக்' செய்யவும்.

மார்ச்சு 2005 முதல் மே 2005 வரையுள்ள எல்லா ஞாயிற்றுக்கிழமைகளும் தானே தொகுக்கப்பட்டு தேர்வு செய்யப்பட்ட நுண்ணறைகளில் தோன்றுவதைப் பார்க்கலாம். (படம் 6.12).

	A	B	C	D	E
1	2	4	8	16	
2	4	8	16	32	
3	8	16	32	64	
4	16	32	64	128	
5	32	64	128	256	
6	64	128	256	512	
7					
8					

படம் 6.11 Fill கட்டளையின் விளைவு

A	B	C	D
Dates of Sundays in March, April and May 2005			
	03/06/2005		
	03/13/2005		
	03/20/2005		
	03/27/2005		
	04/03/2005		
	04/10/2005		
	04/17/2005		
	04/24/2005		
	05/01/2005		
	05/08/2005		
	05/15/2005		
	05/22/2005		
	05/29/2005		

படம் 6.12 Fill கட்டளை மூலம் உருவாக்கப்பட்ட ஞாயிற்றுக்கிழமைகளின் தேதி அட்டவணை.

6.6 நுண்ணுறைப் பார்வையிடல் (Cell Referencing)

மதிப்பெண் அட்டவணைக் கோப்பில், **= C3 + D3 + E3 + F3,** என்ற வாய்பாட்டை **G3** என்ற நுண்ணறையில் **Fill** கட்டளைக் கொண்டு நிரப்பவும். இப்பொழுது **G4–**இல் சுட்டியை வைத்து 'கிளிக்' செய்தால் **G4–**இல் **= C4 + D4+ E4+ F4** என்ற வாய்பாடு தோன்றுவதைப் பார்க்கலாம். இது **StarCalc** அட்டவணைச் செயலியில் உள்ள சிறப்பு அமைப்பாகும்.**StarCalc–**இல் கொடுக்கப்படும் வாய்பாடுகளில் உள்ள நுண்ணறை முகவரிகள் தனித்த ஒரு **(Absolute)** மதிப்பாக எடுத்துக்கொள்ளப்படுவதில்லை. கொடுக்கப்படும் முகவரிகள், ஒப்பிட்டு மதிப்பு **(Relative)** களாகவே எடுத்துக்கொள்ளப்படுகின்றன. மேற்கண்ட எடுத்துக்காட்டில் **G3–**யில் கொடுத்த **= C3 + D3 + E3 + F3** என்ற வாய்பாட்டை**StarCalc** கீழ்க்கண்டவாறு புரிந்து கொண்டு செயல்படுகின்றது.

தற்போது சுட்டி இருக்கின்ற வரிசையில் உள்ள **C,D,E,F** நெடுவரிசை நுண்ணறையில் தேக்கி வைக்கப்பட்டுள்ள எண்களை முறையே இடமிருந்து வலம் நோக்கி ஒன்றன்பின் ஒன்றாகக் கூட்டி விடையை அதே வரிசையில் உள்ள **G** நெடுவரிசை நுண்ணறையில் தேக்கி வைக்க வேண்டும் என்று **StarCalc** புரிந்து கொள்கின்றது. எனவே, சுட்டி **3–**வது வரிசையில் இருக்கும்போது **C3 + D3 + E3 + F3** என்றும், 4–வது வரியில் இருக்கும்போது **C4 + D4 + E4 + F4** என்றும் வாய்பாட்டில் நுண்ணறை முகவரிகள் தானே மாறிவிடுகின்றன. இதனை**Cell Referencing** என்பர்.

இந்த வகை நுண்ணறைப் பார்வையிடல் **(Cell Referencing)** ஒப்பீட்டு நுண்ணறை முகவரி முறை **(Relative Cell Addressing)** என்று அழைக்கப்படுகிறது. **StarCalc–**இல் இந்த முறை, தானே எடுத்துக்கொள்ளும் **(Default –** தற்கோள்) முறையாக அமைக்கப்பட்டுள்ளது.

StarCalc –இல் உள்ள திரும்பக் கணக்கீட்டு **(Recalculation)** வசதியும், ஒப்பீட்டு நுண்ணறை முகவரி **(Relative Cell Addressing)** முறையைத் தழுவியதேயாகும்.

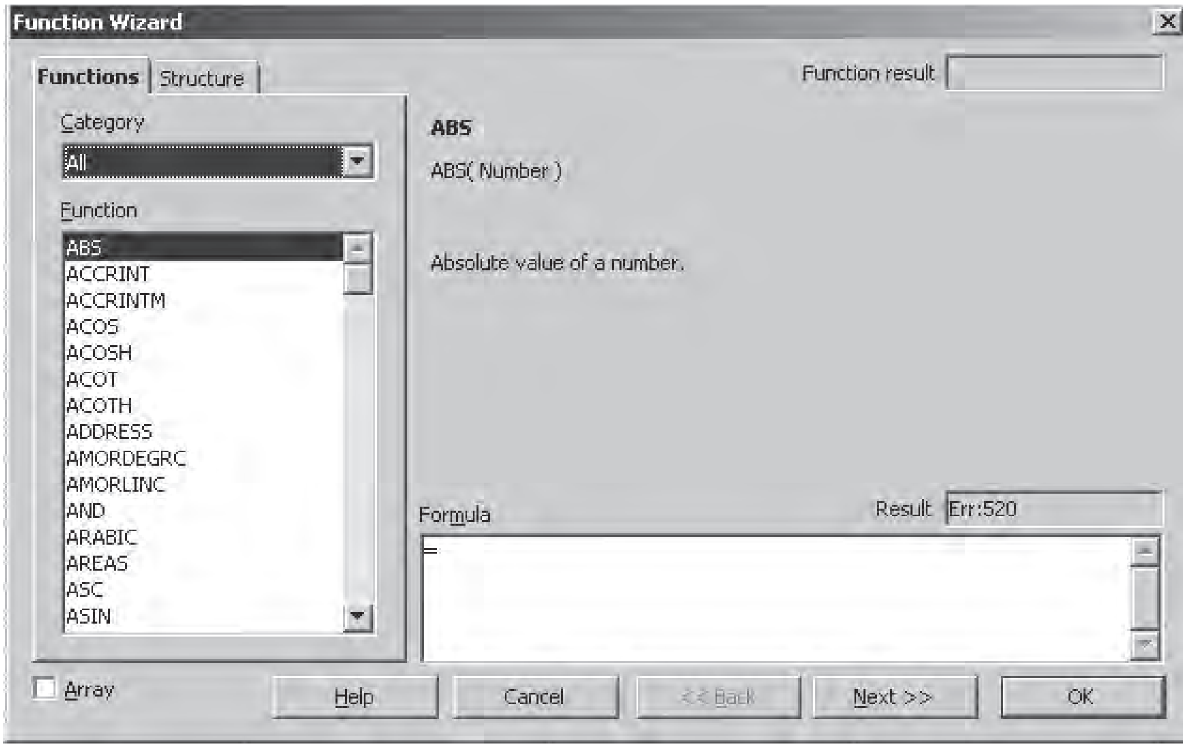
தனித்த நுண்ணறை முகவரி **(Absolute Cell Addressing)** முறையும் **StarCalc–**இல் உள்ளது. வரிசை எண்ணின் முன்பும் நெடுவரிசைத் தலைப்பின் முன்பும் $ என்ற குறியைக் கொடுப்பதன் மூலம் அந்த நுண்ணறை தனித்த முகவரி கொண்டதாக ஆகின்றது.

எடுத்துக்காட்டாக, நுண்ணறை **C4** தனித்த முகவரி கொண்டதாக ஆக்க, **C4** என்று கொடுக்க வேண்டும்.

தனித்த நுண்ணறை முகவரிகள், ஒப்பீட்டு நுண்ணறை முகவரிகள் போல மாறுவதில்லை.

6.7 சார்புகளின் பயன்பாடு (Using Functions)

StarCalc–இல் பல வகையான சார்புகள் **(Functions)** உள்ளன. சார்புகள் என்பன முன்னதாகவே வரையறுக்கப்பட்ட வாய்பாடுகள் ஆகும். இந்த சார்புகள், **StarCalc–**இல் உள்ள**Function Wizard** என்னும் சன்னல் திரை **(Windows)–**யில் உள்ள இழுபட்டி பட்டியலில் **(Pull Down Menu)** உள்ளன. (படம் 6.13).



படம் 6.13 Function AutoPilot சன்னல் திரை

இந்த சார்புகள் பல வகைகளாகப் பிரிக்கப்பட்டுள்ளன. இந்த வகைகள் வகையிழுப்பட்டி பட்டியலில் **(Category Pull - Down Menu)** கொடுக்கப்பட்டுள்ளன.

ஒரு சார்பைத் தேர்ந்தெடுக்க Insert பட்டிக்குள் சென்று **Function** பட்டியைத் தேர்வு செய்ய வேண்டும். உடனே **Function Wizard** உரையாடல் பெட்டி தோன்றும். அதில்**Category** பெட்டியைத் தேர்வு செய்து **All** என்ற பட்டியை ‘கிளிக்’ செய்தால்**Function** பெட்டியில் எல்லா சார்புகளும் தோன்றும். ஒரு குறிப்பிட்ட வகையைத் தேர்ந்தெடுத்தால், அந்த வகையைச் சேர்ந்த தொடர்பான சார்புகள் மட்டும் பட்டியலிடப்பட்டு காட்டப்படும்.

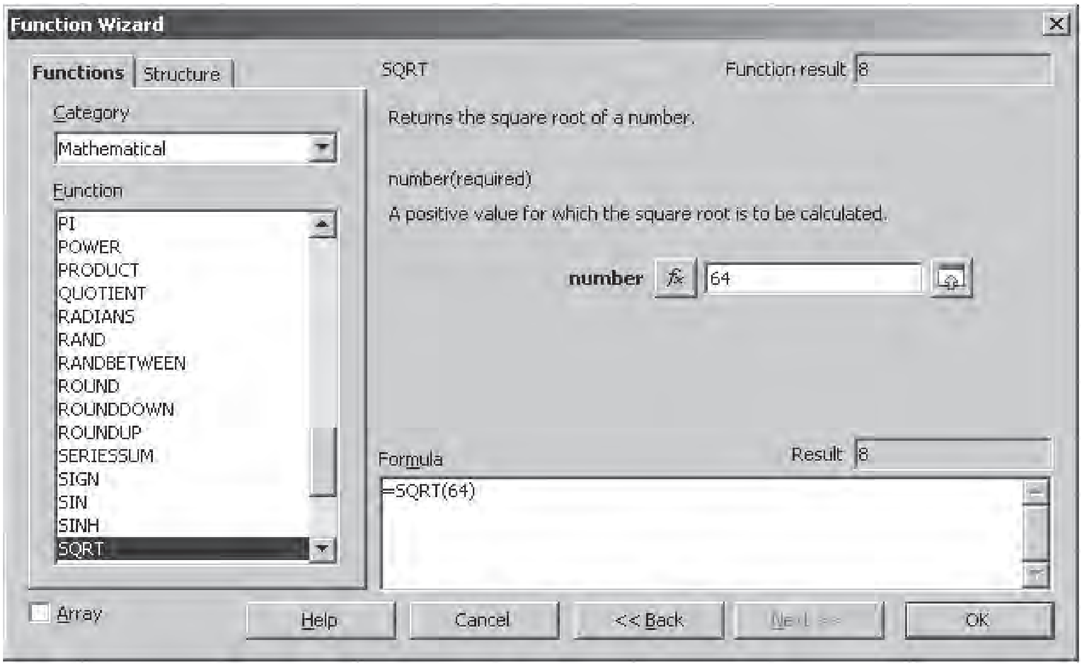
ஒரு குறிப்பிட்ட சார்பைத் தேர்வு செய்ய, **Category** பெட்டியில் அந்த சார்பின் வகையைத் தேர்வு செய். பிறகு தேவையான சார்பைக் கண்டுபிடிக்க, இழு பட்டியை உருளச் செய்.

பிறகு தேவையான சார்பைப் பற்றிய விவரம் அறிய அந்த சார்பின் மீது சுட்டியை வைத்து ஒரு முறை ‘கிளிக்’ செய்க. இந்த சார்பை அட்டவணைக் கோப்பில் பெற இருமுறை ‘கிளிக்’ செய்க.

இந்த **Function Wiard** பணிக்குறி **(icon)**யைப் பயன்படுத்தியும் சார்புகளைத் தேர்வு செய்து அட்டவணையில் பெற முடியும்.

எடுத்துக்காட்டாக, **SQRT** என்ற சார்பை (ஒரு எண்ணின் வர்க்க மூலத்தை **(Square root)** கணிக்கும் சார்பு) அட்டவணைக் கோப்பில் ஒரு நுண்ணறையில் எழுத வேண்டுமானால் கீழ்க்கண்ட வழிமுறையைப் பின்பற்றவும். எந்த நுண்ணறையில் **SQRT** என்ற சார்பை எழுத வேண்டுமோ அந்த நுண்ணறையைத் தேர்வு செய்து **Function Auto Pilot** என்ற பணிக்குறியை**(icon)** ‘கிளிக்’ செய். பிறகு **Category** பெட்டியில் **Mathematical** என்ற வகையைத் தேர்வு செய் இப்பொழுது இழுபட்டியை **(Pull Down Menu)** உருளச் செய்து**SQRT** என்ற சார்பைத் தேர்வு செய்து இருமுறை ‘கிளிக்’ செய். உடனே**Auto Pilot** சார்பு ஒரு சிறு

விளக்கத்தை காட்டி, எண்ணை உள்ளிடத் தூண்டும். அப்பொழுது எந்த எண்ணின் வர்க்க மூலத்தைக் கண்டுபிடிக்க வேண்டுமோ அந்த எண்ணை (இங்கு 64 என்ற எண்ணைக்) கொடு. பிறகு **OK** என்ற பட்டியை 'கிளிக்' செய். உடனே **SQRT** சார்பு 64 என்ற எண்ணின் வர்க்க மூலத்தைக் கண்டுபிடித்து 8 என்ற எண்ணை விடையாக நுண்ணறையில் காட்டுவதை படம் 6.14 –இல் காணலாம்.



படம் 6.14 Auto Pilot சார்பு நுண்ணறையில் எண்ணை உள்ளிடத்தூண்டும் திரை.

Σ **SUM** என்ற சார்பு மிகுதியாகப் பயன்படும் ஒரு சார்பாகும். இந்த சார்பு, கொடுக்கப்பட்ட எண்களின் கூட்டுத் தொகையை கணக்கிட்டு கொடுக்கின்றது. இந்த சார்பைப் பயன்படுத்த **Function Auto Pilot** பணிக்குறியைக் **(icon)** 'கிளிக்' செய்து பிறகு **SUM** என்ற சார்பை தேர்வு செய்து பெறலாம். அல்லது நேரடியாக வாய்பாட்டுப் பட்டையில் உள்ள **SUM** என்ற பணிக்குறியை 'கிளிக்' செய்தும் பெறலாம். **StarCalc** கூட்டல் செய்ய வேண்டிய நுண்ணறைகளின் பரப்பை வரையறுத்துக் கூற வழிசெய்கின்றது. நுண்ணறைகளின் பரப்பை வரையறுக்க **Enter** என்ற பொத்தானை அழுத்தி அல்லது **Backspace** பொத்தானை அழுத்தி நுண்ணறைகளின் பரப்பை, வேண்டிய அளவிற்குக் கொடுக்கலாம்.

எடுத்துக்காட்டாக, மதிப்பெண் அட்டவணைக் கோப்பில் மதிப்பெண்களின் கூட்டுத் தொகையை கணக்கிட கீழ்க்கண்ட வழிமுறையைச் செய்க.

- சுட்டியை **G3** யில் இட்டு 'கிளிக்' செய்க.
- சார்புப் பட்டையில் உள்ள **SUM** என்ற பணிக்குறியைக் **(icon)** 'கிளிக்' செய்க. உடனே **= SUM ( )** என்ற குறித்தோன்றும். அடைப்புக்குள் **C3 : F3** என்று உள்ளிடவும். பிறகு **= SUM (C3: F3)** என்ற வாய்பாடு நுண்ணறையில் தெரியும்.
- **C3 : F3** என்ற **C3** முதல் **F3** வரை உள்ள நுண்ணறைப் பரப்பை, ஏற்க **Enter** பொத்தானை அழுத்துக. இப்பொழுது விடையாகிய 332 என்ற கூட்டுத்தொகை **G3** என்ற நுண்ணறையில் தோன்றும்.

செய்து பார்

1. **Marks** என்ற அட்டவணைக் கோப்பை திற.
2. ஒரு வாய்பாட்டைப் பயன்படுத்தி கூட்டுத்தொகையைக் கணக்கிட்டு **G3** யில் எழுது.
3. **Fill** கட்டளையைப் பயன்படுத்தி வாய்பாட்டை **G4** முதல் **G7** வரை **(G4 : G7)** உள்ள நுண்ணறைகளில் எழுது.
4. சேமிக்காமல் அட்டவணைக் கோப்பை மூடு.
5. அட்டவணைக் கோப்பை மீண்டும் திற. பிறகு **SUM** என்ற சார்பைப் பயன்படுத்தி **G3** யில் கூட்டுத்தொகையைக் கணக்கிட்டு எழுது.
6. **Fill** கட்டளையைப் பயன்படுத்தி வாய்பாட்டை **G4 : G7** நுண்ணறைகளில் எழுது.
7. அட்டவணைக் கோப்பை சேமி.
8. **H3–**யில் சராசரி மதிப்பெண்ணைக் கணக்கிட வாய்பாடு எழுது.
9. **Fill** கட்டளையைப் பயன்படுத்தி வாய்ப்பாட்டை **H4 : H7** நுண்ணறைகளில் எழுது.
10. அட்டவணைக் கோப்பை சேமிக்காமல் மூடு.
11. அட்டவணைக் கோப்பை மீண்டும் திற. பிறகு **Average** சார்பைப் பயன்படுத்தி சராசரி மதிப்பெண்ணைக் கணக்கிட்டு **H3** யில் எழுது.
12. **Fill** கட்டளையைப் பயன்படுத்தி வாய்பாட்டை**H4 : H7** நுண்ணறைகளில் எழுது.
13. அட்டவணைக் கோப்பை சேமித்து மூடு.

6.8 தேதிக் கணக்கீடு

தேதிக் கணக்கீடுகளை கையால் செய்யும் பொழுது நாம் ஒவ்வொரு மாதத்தின் சரியான நாட்களையும், ஆண்டு லீப் ஆண்டா இல்லையா போன்ற தகவல்களையும் நினைவில் வைத்துக் கொள்ள வேண்டியுள்ளது. இதனை கணிப்பொறியில் அட்டவணைச் செயலி முலம் செயல்படுத்துவது எளிதாகும். அட்டவணைச் செயலி மூலம் நீங்கள் தேதியுடன் நாட்களைக் கூட்டி புது தேதியைக் கண்டு பிடிக்கலாம். இரு தேதிகளை கழித்து இடைப்பட்ட நாட்களைக் கணக்கிடலாம். இவை போன்ற பல கணக்கீடுகளையும் வடிவமைப்புகளையும் **(Format)** செய்யலாம்.

எடுத்துக்காட்டாக 03/04/00 என்ற தேதியை நுண்ணறை **A2–**இல் கொடு. தேதியைக் கொடுக்கும்போது முதலில் மாதத்தையும் பிறகு நாள் அதன் பிறகு ஆண்டு என்ற வரிசையில் கொடுக்க வேண்டும் என்பதை நினைவில் கொள்க. இப்பொழுது, 79 நாட்களுக்குப் பிறகு என்ன தேதி என்று அறிய விரும்புவதாகக் கொள்வோம். இதனைச் செய்ய வாய்பாடு **= A2 +79** என்று இன்னொரு நுண்ணறை **A4–**இல் கொடுக்கவும். தேதி 05/22/00 என்று நுண்ணறை **A4–**இல் தோன்றுவதைக் காணலாம்.

இன்னொரு எடுத்துக்காட்டாக, 05/10/00, 12/08/00 என்று இரு தேதிகளுக்கு இடைப்பட்ட நாட்களைக் கணக்கிட விரும்புவதாகக் கொள்வோம். இதனைச் செய்ய இந்த தேதிகளை

ஏதேனும் இரு நுண்ணறைகளில் **(B2,B3**என்க) கொடுக்கவும். பிறகு **= B2-B3** என்ற வாய்பாட்டை நுண்ணறை **B4**–இல் கொடுக்கவும். விடை **10746** நாட்கள் என்று நுண்ணறை **B4**–இல் தோன்றுவதைக் காணலாம்.

6.9 அட்டவணைத் தாளை வடிவமைத்தல் (Formatting the Worksheet)

முன் அதிகாரங்களில் **Star Writer** –இல் உள்ள வடிவமைப்பு வகைகள் **(Formatting Options)** பற்றி படித்தோம். அவற்றின் பல வகைகளை **StarCalc**–லும் பயன்படுத்தலாம். அவற்றுடன், **StarCalc** எண்களை வடிவமைக்க பல வகைகளைக் கொடுக்கின்றது. நுண்ணறையில் உள்ள தரவுகளை வடிவமைக்கும் போது, அவை திரையில் தோன்றும் வடிவம்தான் மாறுகிறதே தவிர, தரவின் மதிப்பு மாறுவதில்லை என்பதை நினைவில் கொள்க.

நுண்ணறைத் தரவுகளை வடிவமைக்கு முன் நுண்ணறைகளைத் தேர்ந்து எடுக்க வேண்டும். இது பற்றி முன்பே படித்துள்ளோம்.

ஒரு நுண்ணறையைத் தேர்வு செய்ய அந்த நுண்ணறையின் மீது ‘கிளிக்’ செய்ய வேண்டும். நுண்ணறைகளின் ஒரு தொகுப்பை தேர்வு செய்ய, முதல் நுண்ணறை மீது ‘கிளிக்’ செய்து, பிறகு கடைசி நுண்ணறை வரை சுட்டியை இழுத்துச் செல்ல வேண்டும்.

ஒரு வரிசை முழுவதையும் அல்லது ஒரு நெடுவரிசை முழுவதையும் தேர்வு செய்ய வேண்டுமானால், தேர்வு செய்த வரிசை எண் அல்லது நெடுவரிசைப் பெயரில் சுட்டியை வைத்து ‘கிளிக்’ செய்ய வேண்டும்.

ஒரு வரிசைக்கு மேல் அல்லது ஒரு நெடுவரிசைக்கு மேல் தேர்வு செய்ய வேண்டுமானல், தேர்வு செய்த முதல் வரிசை எண் அல்லது முதல் நெடுவரிசைப் பெயரில் ‘கிளிக்’ செய்து தேவைப்பட்ட கடைசி வரிசை எண் அல்லது கடைசி நெடுவரிசைப் பெயர் வரை சுட்டியை இழுத்துச் செல்ல வேண்டும்.

StarCalc–இல் உள்ள சில வடிவமைப்பு வகைகள் கீழே கொடுக்கப்பட்டுள்ளன.

இது தடிப்புப் பணிக்குறி **(Bold Icon)**. இதனைப் பயன்படுத்தி தரவுகளை தடித்த எண் எழுத்துக்களில் காட்டலாம். இதனைப் பயன்படுத்த இந்த பணிக்குறி மீது சுட்டியை வைத்து ‘கிளிக்’ செய்யவும்.

இந்த பணிக்குறி தரவுகளை சாய்ந்த வடிவில் காட்டப் பயன்படும்.

இது அப்பொழுது செயல்பாட்டில் இருக்கும் நுண்ணறையின் **(Cell)** தரவுகளைக் கீழ்க் கோடிட்டு **(Underline)** காட்ட பயன்படும்.

இது தரவுகளின் எண்/ எழுத்துக்களின் வடிவுகளை **(Font)** களை மாற்றியமைக்க பயன்படும். இதில் இழு பட்டி **(Pull Down Menu)** மூலம் தேவையான வடிவுவகையை **(Font)** தேர்ந்து ‘கிளிக்’ செய்ய வேண்டும்.

இது தரவின் வடிவு அளவை **(Font Size)** மாற்றி அமைக்கப் பயன்படும்.

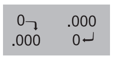
இது தரவின் வடிவின் நிறத்தை **(Font Colour)** மாற்றி அமைக்கப் பயன்படும்.

இவை தரவுகளை, இடப்புற வரிசைப்பாடு **(Left Alignment)**, வலப்புற வரிசைப்பாடு **(Right Alignment)**, மையவரிசைப்பாடு **(Centre Alignment)** போன்ற ஒழுங்கு முறைகளைச் செய்ய உதவுகிறது.

இது பணத்தைக் குறிக்கும் வடிவமைப்பு **(Number Format : Currency)** பணிக்குறி, இந்த பணிக்குறியை 'கிளிக்' செய்தால் நுண்ணறையில் உள்ள தரவுகள் தேர்வு செய்து காட்டினால் பணக்குறியீட்டுடன் திரையில் தோன்றும். எடுத்துக்காட்டாக $ அல்லது Rs. போன்ற குறியீடுகள் தரவிற்கு முன் தோன்றும்.

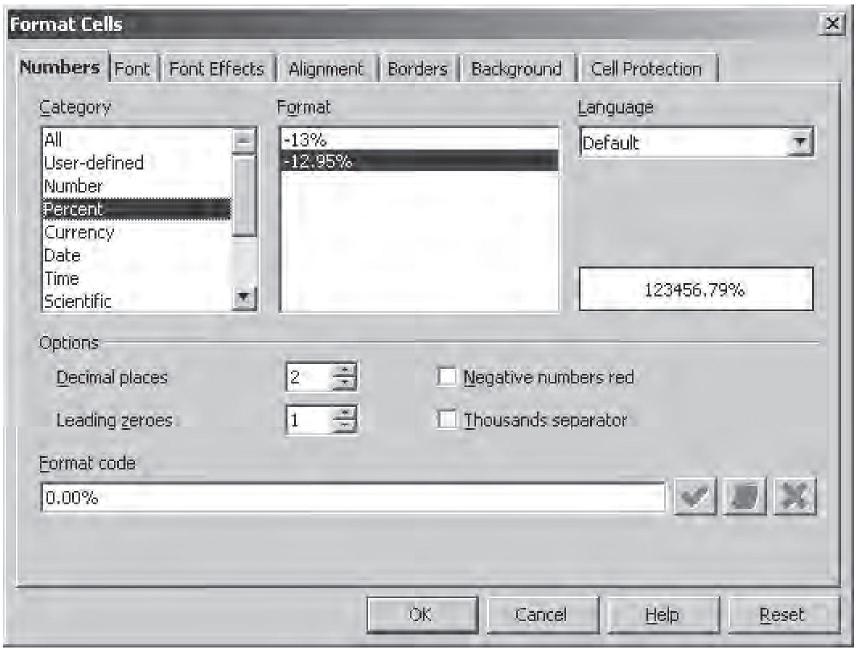
இது விழுக்காட்டைக் குறிக்கும் எண் வடிவமைப்புப் பணிக்குறி **(Number Format : Percent)** ஆகும். இந்த பணிக்குறியை 'கிளிக்' செய்தால் நுண்ணறைத் தரவு விழுக்காட்டில் **(Percentage)** இருதசமப்புள்ளி எண்களுடன் கொடுக்கப்படும்.

இது எண்களை இயல்பான வடிவில் காட்டும் பணிக்குறி **(Number Format : Default icon)** ஆகும்.



இது எண்களின் தசம எண் அளவைக் கூட்டவோ, குறைக்கவோ பயன்படும் பணிக்குறிகளாகும். **(Number Format : Add Decimal, Number Format : Delete Decimal Icons)**

மேற்கூறிய நுண்ணறைத் தரவு அமைப்புகளை, வடிவமைப்புப் **(Format)** பட்டி மூலமும் செயற் படுத்தலாம். இதனைச் செய்ய **Format → Cells** பட்டிகளை 'கிளிக்' செய்ய வேண்டும். அப்பொழுது **Format Cells** உரையாடல் பெட்டி (**Dialog Box)** திறக்கும் (படம் 6. 15). உரையாடல் பெட்டியின் மேற்பகுதியில் உள்ள பணிப்பட்டியில் **Numbers** அல்லது **Alignment** போன்ற தேவைப்பட்ட பணிப்பட்டியை தேர்வு செய்து 'கிளிக்' செய்து செயல்படலாம். 'கிளிக்' செய்தவுடன் வடிவமைப்பு வகைகள் அவற்றின் முன்மாதிரி திரையுடன் தோன்றும்.

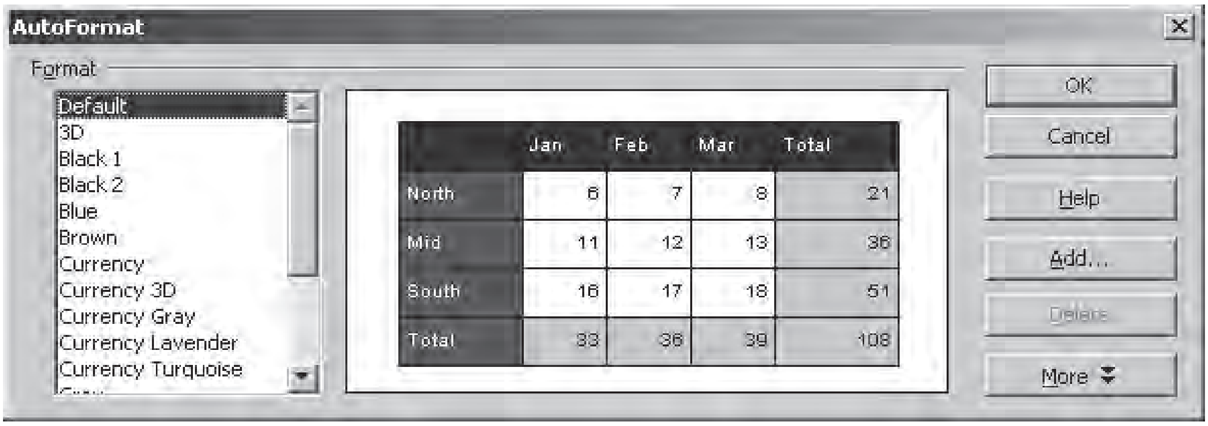


படம் 6.15 Format Cells உரையாடல் பெட்டி

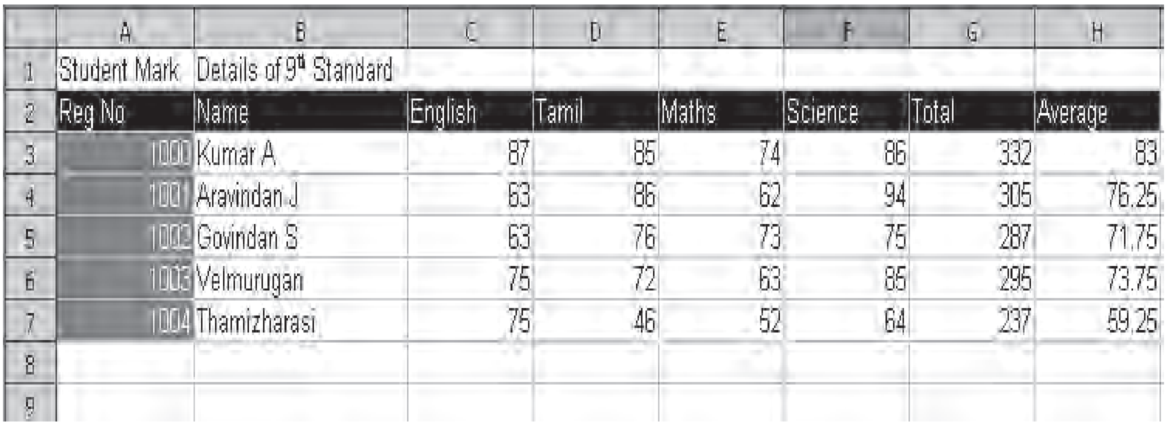
6.9.1 தானே வடிவமைக்கவல்ல அட்டவணைத்தாள் (Auto Format Sheet)

StarCalc, அட்டவணைக் தாட்களின் பாணி **(Style),** வண்ணம் **(Colour),** ஆகியவற்றை முன்பே வரையறுக்கப்பட்டதற்கேற்ப தானே உருவாக்கிக் கொள்ளும் வசதியைக் கொண்டுள்ளது. எடுத்துக்காட்டாக, மாணவர்களின் மதிப்பெண்அட்டவணைத்தாளை மேலே கூறிய, தானே வடிவமைக்கும் வசதியைக் கொண்டு, வடிவமைப்பதாகக் கொள்வோம். இதனைச் செய்ய கீழ்க்கண்டவாறு செயல்படு:

- மதிப்பெண் அட்டவணைத்தாளில் உள்ள நுண்ணறைகள் **A1** முதல் **H7** வரை தேர்வு செய்.
- **Format** பட்டிக்குள் சென்று **Auto Format** பட்டியை ‘கிளிக்’ செய்.
- **Auto Format** உரையாடல் தோன்றும் (படம் 6.16). இதில் தேவையான பாணி**(Style),**வண்ணம் **(Colour)** ஆகியவற்றைக் குறி.
- **Default** வடிவமைப்பைத் தேர்வு செய்து கிடைத்த அட்டவணைத்தாள் படம் 6.17 –இல் காட்டப்பட்டுள்ளது.



படம் 6.16 **Auto Format** உரையாடல் பெட்டி



	A	B	C	D	E	F	G	H
1	Student Mark	Details of 9th Standard						
2	Reg No	Name	English	Tamil	Maths	Science	Total	Average
3	1000	Kumar A	87	85	74	86	332	83
4	1001	Aravindan J	63	86	62	94	305	76.25
5	1002	Govindan S	63	76	73	75	287	71.75
6	1003	Velmurugan	75	72	63	85	295	73.75
7	1004	Thamizharasi	75	46	52	64	237	59.25
8								
9								

படம் 6.17 **Auto Format** மூலம் வடிவமைத்த அட்டவணைத்தாள்.

6.10 அட்டவணைத்தாளின் நெடுவரிசை அகலத்தையும், வரிசையின் உயரத்தையும் மாற்றுதல்.

அட்டவணைத்தாளில் தரவுகள் கொடுக்கும்போது நெடுவரிசையின் அகலம் போதாமல் போகும் நிலையை அடிக்கடி சந்தித்து இருப்பீர்கள். நெடுவரிசையின் அகலத்தை அல்லது வரிசையின் உயரத்தை மாற்றி அமைக்க **StarCalc** –இல் வழியுள்ளது. நெடுவரிசையின் அகலத்தை

மாற்ற, **Format → Column→ Width** பட்டிகளை 'கிளிக்' செய்து தோன்றும் உரையாடல் பெட்டியில் (படம் 6.18) புதிய, தேவையான அகலத்தைக் கொடுத்து **OK** பட்டியை 'கிளிக்' செய்யவும். இதனை வேறு வழியில் கூட செய்யலாம். எந்த நெடுவரிசையின் அகலத்தை மாற்ற விரும்புகின்றோமோ அந்த நெடுவரிசையை, அடுத்த நெடுவரிசையிலிருந்து பிரிக்கும் கோட்டிற்கு சுட்டியை எடுத்துச் செல்லவும். சுட்டெலியின் சுட்டி இரு தலை அம்பாக மாறுவதைக் காணலாம். அந்த நிலையில் 'கிளிக்' செய்து சுட்டியை இடப்புறம் அல்லது வலப்புறம் இழுத்து அகலத்தை குறைக்கவோ அல்லது கூட்டவோ செய்யலாம்.

வரிசையின் உயரத்தை மாற்ற, எந்த வரிசையின் உயரத்தை மாற்ற விரும்புகின்றோமோ, அந்த வரிசையை தேர்வு செய்ய வேண்டும். பிறகு **Format→ Row →Height** பொத்தான்களை கிளிக் செய்து பிறகு, தேவையான புதிய வரிசையின் உயர அளவை கொடுத்து **OK** என்ற பட்டியை 'கிளிக்' செய்யவும். (படம் 6. 19). இதனை வேறு வழியிலும் செய்யலாம். எந்த வரிசையின் உயரத்தை மாற்ற விரும்புகின்றோமோ, அந்த வரிசையை, அடுத்த வரிசையிலிருந்து பிரிக்கும் கோட்டிற்கு, சுட்டியை எடுத்துச்சென்று 'கிளிக்' செய்து மேலே அல்லது கீழே இழுத்து வரிசையின் உயரத்தை குறைக்கவோ அல்லது கூட்டவோ செய்யலாம்.

படம் 6 18 நெடுவரிசை அகலத்தை மாற்றும் திரை

படம் 6. 19 வரிசையின் உயத்தை மாற்றும் திரை

செய்து பார்

மாணவர் மதிப்பெண் அட்டவணைத் தாளை கீழ்க்கண்டவாறு வடிவமைக்கவும்.

1. எல்லாத் தலைப்புகளையும் தடித்த எழுத்துக்களில் வடிவமைக்கவும்.

2. தலைப்புகளின் வடிவுவகையை **(Font Type)**, வடிவு அளவை **(Font Size)** வண்ணத்தை மாற்றி அமை.

3. **Average** என்ற தலைப்பைக் கொண்ட நெடுவரிசையின் வடிவமைப்பை, எண்கள் இரண்டு தசம இடங்களோடு வருமாறு மாற்றி அமை.

4. **Auto Format** வசதியைப் பயன்படுத்தி அட்டவணைத்தாளின் பாணி, வண்ணம் ஆகியவற்றை மாற்றி அமை.

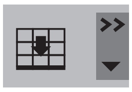
5. வரிசை மற்றும் நெடுவரிசை அகலத்தை தேவைப்பட்ட இடங்களில் எல்லாம் மாற்றி அமை.

6.11 அட்டவணைத் தாளில் நுண்ணுறைகள், வரிசைகள், நெடுவரிசைகள் ஆகியவற்றை சேர்த்தல்

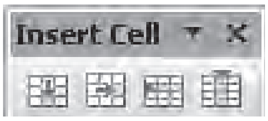
ஒரு அட்டவணைத் தாளை உருவமைத்த பிறகு, அட்டவணையில் தரவுகளை கொடுத்து ஒரு குறிப்பிட்ட அட்டவணையை உருவாக்கும் போது, புதிய வரிசை அல்லது நெடுவரிசையை சேர்க்க வேண்டிய நிலையை அடிக்கடி சந்திக்கின்றோம். எடுத்துக்காட்டாக, புதிய ஒரு பாடத்தை மதிப்பெண் அட்டவணைத்தாளில் சேர்க்க நேரலாம். புதிய ஒன்று அல்லது அதற்கு மேற்பட்ட நுண்ணறைகள், வரிசைகள், அல்லது நெடுவரிசைகளை, ஏற்கனவே உருவமைத்த அட்டவணைத் தாளில் சேர்க்க **StarCalc** வழி செய்கின்றது.

ஒரு காலி நுண்ணறையை அல்லது காலி வரிசையை அல்லது காலி நெடுவரிசையை, ஏற்கனவே உருவமைத்த அட்டவணைத்தாளில் சேர்க்க, கீழ்கண்ட வழிமுறையை பின்பற்றவும்:

- பணிக்கருவிப் பட்டைப் பட்டியலில் **(Tool Bar)** உள்ள insert என்ற பணிக்குறியை



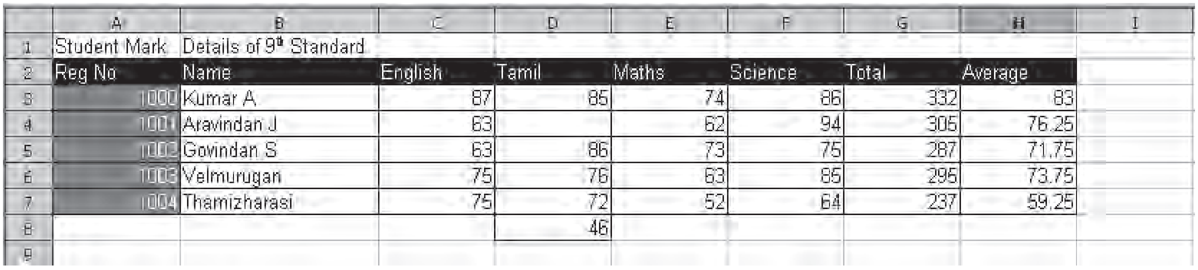
‘கிளிக்’ செய்யவும்.



நான்கு பணிக்குறிகளைக் கொண்ட ஒரு தற்காலிக பணிக்கருவிப் பட்டை **(Floating Tool Bar)** தோன்றும். அந்த பணிக்குறிகளாவன:

Insert Cells Down
Insert Cells Right
Insert Rows
Insert Columns

- ஒரு காலி நுண்ணறையை ஒரு நெடுவரிசையில் சேர்க்க, சுட்டியை புது நுண்ணறையை சேர்க்க விரும்பும் இடத்தில் வைத்து தேர்வு **(Select)** செய். பிறகு Insert Cells Down என்ற பணிக்குறியை ‘கிளிக்’ செய்யவும். எடுத்துக்காட்டாக மதிப்பெண் அட்டவணைத் தாளில் புதிதாக D4 என்ற காலி நுண்ணறையை சேர்ப்பதாகவும்; D4:D7 நுண்ணறைகளை கீழே நகர்த்துவதாகவும் கொள்க. இதனைச் செய்ய D4 என்ற நுண்ணறையை தேர்வு செய்து என்ற இந்த பணிக்குறியை ‘கிளிக்’ செய்யவும். புதிய நுண்ணறை உருவாக்கப்பட்ட அட்டவணைத் திரையை படம் 6.20 –இல் காணலாம்.



	A	B	C	D	E	F	G	H	I
1	Student Mark	Details of 9th Standard							
2	Reg No	Name	English	Tamil	Maths	Science	Total	Average	
3	1000	Kumar A	87	85	74	86	332	83	
4	1001	Aravindan J	63		62	94	305	76.25	
5	1002	Govindan S	63	86	73	75	287	71.75	
6	1003	Velmurugan	75	76	63	85	295	73.75	
7	1004	Thamizharasi	75	72	52	64	237	59.25	
8				46					
9									

படம் 6.20 நுண்ணறை உருவாக்கப்பட்ட அட்டவணைத் தாள்

- ஒரு நுண்ணறையில் உள்ள தரவை வலப்புறம் நகர்த்தி விட்டு ஒரு காலி நுண்ணறையை உருவாக்க, அந்த குறிப்பிட்ட நுண்ணறையை தேர்வு செய்து **Insert Cells Right** என்ற பணிக்குறியை ‘கிளிக்’ செய்ய வேண்டும். எடுத்துக்காட்டாக, மதிப்பெண் அட்டவணைத் தாளின் ஐந்தாவது வரிசையில் உள்ள தரவுகளை **C5 : F5** நுண்ணறைகளில் இருந்து **D5 : G5** நுண்ணறைகளுக்கு மாற்ற விரும்புவதாகக் கொள்க. இதனைச் செய்ய நுண்ணறை **C5**ஐ தேர்வு செய்து என்ற பணிக்குறியை ‘கிளிக்’ செய்யவும். மாற்றப்பட்ட அட்டவணைத் திரையை படம் **6.21** இல் காண்க.

	A	B	C	D	E	F	G	H	I
1	Student Mark	Details of 9th Standard							
2	Reg No	Name	English	Tamil	Maths	Science	Total	Average	
3	1000	Kumar A	87	85	74	86	332	83	
4	1001	Aravindan J	63	86	62	94	305	76.25	
5	1002	Govindan S		63	76	73	75	287	71.75
6	1003	Velmurugan	75	72	63	85	295	73.75	
7	1004	Thamizharasi	75	46	52	64	237	59.25	
8									
9									

படம் 6.21 வலப்புறம் நகர்த்தப்பட்ட நுண்ணறைகளைக் கொண்ட அட்டவணைத்தாள்

- அட்டவணைத் தாளில் ஒரு காலி வரிசையைச் சேர்க்க, எங்கு புதிய வரிசையைச் சேர்க்க விரும்புகின்றோமோ அந்த வரிசையை தேர்வு செய்து **Insert Rows** என்ற பணிக்குறியை ‘கிளிக்’ செய்யவும். எடுத்துக்காட்டாக மதிப்பெண் அட்டவணைத் தாளின் வரிசைகள் நான்கிற்கும் ஐந்திற்கும் இடையில் ஒரு புது வரிசையை உருவாக்குவதாகக் கொள்வோம். இதனைச் செய்ய வரிசை ஐந்தை தேர்வு செய்து என்ற பணிக்குறியை ‘கிளிக்’ செய்யவும். இப்படி உருவாக்கப்பட்ட அட்டவணைத் திரையை படம் **6.22** இல் காண்க.

	A	B	C	D	E	F	G	H	I
1	Student Mark	Details of 9th Standard							
2	Reg No	Name	English	Tamil	Maths	Science	Total	Average	
3	1000	Kumar A	87	85	74	86	332	83	
4	1001	Aravindan J	63	86	62	94	305	76.25	
5									
6	1002	Govindan S	63	76	73	75	287	71.75	
7	1003	Velmurugan	75	72	63	85	295	73.75	
8	1004	Thamizharasi	75	46	52	64	237	59.25	
9									

படம் 6.22 புது வரிசைச் சேர்க்கப்பட்ட அட்டவணைத்தாள்

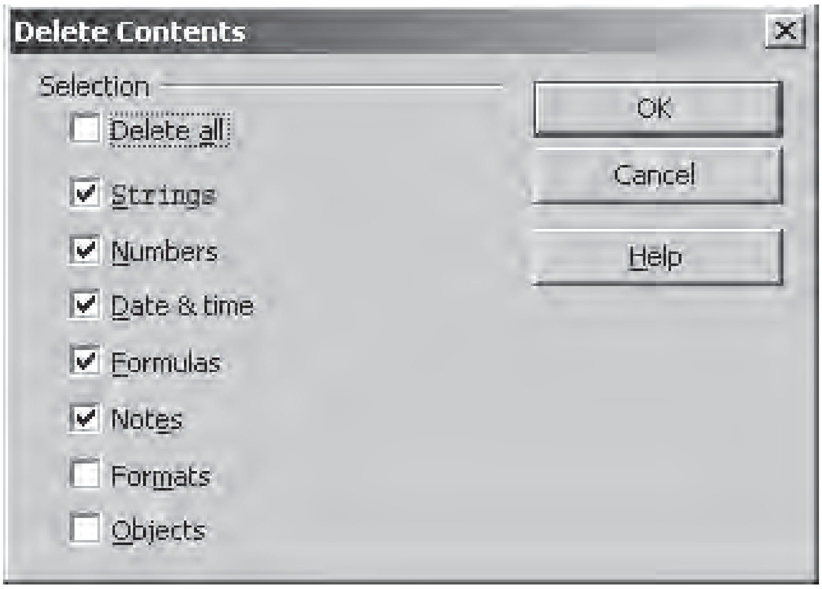
- அட்டவணைத்தாளில் ஒரு நெடுவரிசையைச் சேர்க்க, எந்த இடத்தில் நெடுவரிசையைச் சேர்க்க விரும்புகின்றோமோ, அந்த இடத்திலுள்ள நெடுவரிசையைத் தேர்வு செய்து **Insert Columns** என்ற பணிக்குறியை ‘கிளிக்’ செய்ய வேண்டும். எடுத்துக்காட்டாக மதிப்பெண் அட்டவணைத் தாளில் நெடுவரிசைகள் **E**க்கும் **F**க்கும் இடையில் ஒரு புதிய நெடுவரிசையை உருவாக்குவதாகக் கொள்வோம். இதனைச் செய்ய நெடுவரிசை **F**ஐ தேர்வு செய்து பணிக்குறியை ‘கிளிக்’ செய்யவும். இவ்வாறு மாற்றப்பட்ட அட்டவணைத்தாளைப் படம் **6.23** இல் காண்க.

	A	B	C	D	E	F	G	H	I
1	Student Mark	Details of 9th Standard							
2	Reg No	Name	English	Tamil	Maths		Science	Total	Average
3	1000	Kumar A	87	85	74		86	332	83
4	1001	Aravindan J	63	86	62		94	305	76.25
5									
6	1002	Govindan S	63	76	73		75	287	71.75
7	1003	Velmurugan	75	72	63		85	295	73.75
8	1004	Thamizharasi	75	46	52		64	237	59.25
9									

படம் 6.23 புது நெடுவரிசை சேர்க்கப்பட்ட அட்டவணைத்தாள்

6.12 அட்டவணைத் தாளில் இருந்து நுண்ணுறைகள், வரிசைகள், நெடுவரிசைகள் ஆகியவற்றை நீக்குதல்.

நுண்ணறைகள், வரிசைகள், நெடுவரிசைகள் ஆகியவற்றில் எதை அட்டவணையிலிருந்து நீக்க வேண்டும் என்றாலும் பயன்படும் வழிமுறைகள் ஒன்றே. **Delete Contents** நுண்ணறை அல்லது நுண்ணறை வரிசையை நீக்க வழி செய்கிறது. நீக்க வேண்டிய நுண்ணறை அல்லது வரிசை அல்லது நெடுவரிசையைத் தேர்வு செய்து பிறகு **Edit→Delete Contents** என்ற பட்டிகளை கிளிக் செய்து **OK** என்ற பட்டியை 'கிளிக்' செய்தால் தேர்வு செய்த நுண்ணறை, வரிசை அல்லது நெடுவரிசை நீக்கப்படும். **Delete All** என்ற பட்டியை 'கிளிக்' செய்து, **OK** பட்டியை கிளிக் செய்தால் தேர்வு செய்த அத்தனையும் நீக்கப்படும்.



படம் 6.24 தரவுகளை நீக்கல்

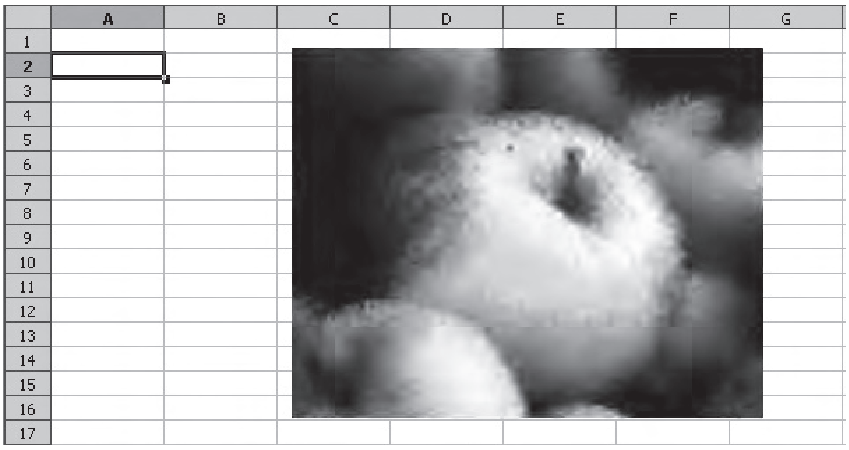
தேர்வு செய்யப்பட்ட ஒரு நுண்ணறையை மட்டும் நீக்க, **Delete Cells** என்ற பட்டியைத் தேர்வு செய்யவும். உடனே படம் 6.25 இல் காட்டியுள்ள சன்னல்திரை தோன்றும். அதில் நுண்ணறைகள் நீக்கப்பட்ட பின் அட்டவணை எப்படி அமைய வேண்டும் என்பதற்கான மாற்று வழிகள் காணப்படும். அதில் ஒன்றைத் தேர்வு செய்து செயல்படுத்துக. **Shift Cells Up** என்ற பட்டியைத் தேர்வு செய்து OK பட்டியை கிளிக் செய்தால், காலி செய்யப்பட்ட நுண்ணறை அதற்குக் கீழே உள்ள நுண்ணறையால் நிரப்பப்படும்.**Shift Cells Left** என்ற பட்டியைத் தேர்வு செய்தால், காலி செய்யப்பட்ட நுண்ணறை அதற்கு வலப்புறம் உள்ள நுண்ணுறையால் நிரப்பப்படும்.

படம் 6.25 நுண்ணறையை நீக்கல்

6.13 அட்டவணைத் தாளில் படங்களையும் சிறப்புக் குறியீடுகளையும் சேர்த்தல்.

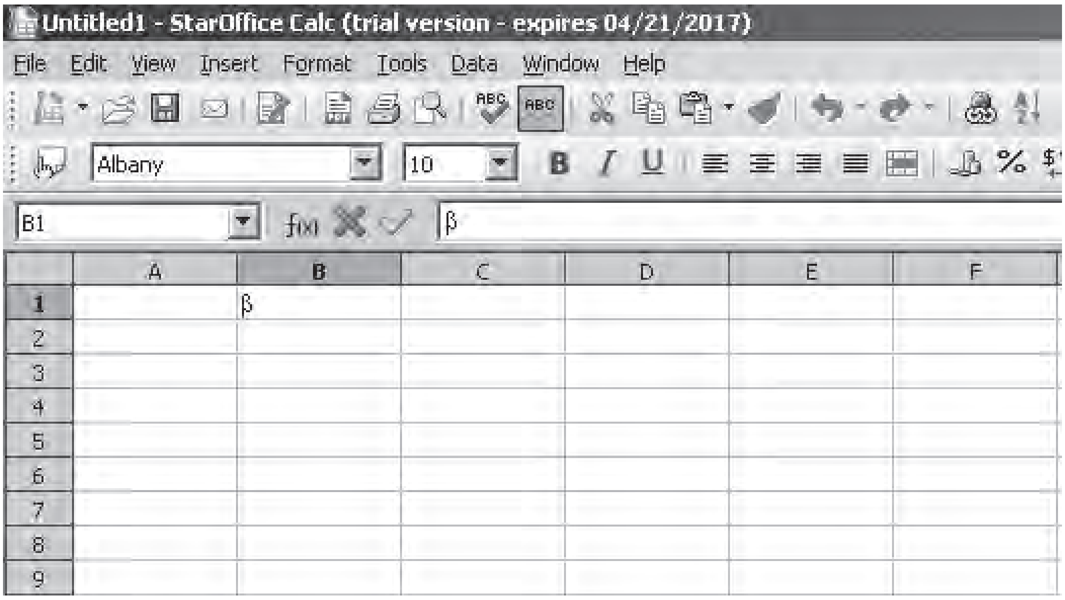
அட்டவணைத் தாளில் படங்கள், α, β போன்ற சிறப்புக் குறியீடுகள் இவற்றைச் சேர்க்க **StarCalc** வழிவகை செய்கின்றது. இவற்றுக்கான பணிக்குறிகள் முதன்மை பணிக்கருவிப் பட்டையில் உள்ளன. படத்தையோ சிறப்புக்குறியீடுகளையோ அட்டவணையில் சேர்க்க கீழ்க்கண்ட வழிமுறைகளை பின்பற்றவும்:

- நுண்ணறைச் சுட்டியை ஏதாவது ஒரு நுண்ணறைக்குக் கொண்டு செல்லவும் (**B2**என்க)
- முதன்மை பணிக்கருவிப் பட்டையில் **Insert →Picture → From File** என்ற பணிக்குறிகளைத் தேர்வு செய்க.
- **Insert Picture** பட்டித் தோன்றும்.
- ஒரு படத்தை சேர்க்க **Insert Picture** பணிக்குறியை 'கிளிக்' செய்யவும். உடனே **Insert Picture** என்ற உரையாடல் பெட்டி **(Dialogue Box)** தோன்றும். உரையாடல் பெட்டியில் உள்ள File பட்டியில் படம் இருக்கின்ற கோப்பின் பெயரையும், கோப்பகப் **(Directory)** பெயரையும் **(Path)** கொடுக்கவும். (சேர்க்க வேண்டிய படத்தை **Star Office** இன் **Gallery Directory** இல் இருந்து நேரடியாகவும் தேர்வு செய்துக் கொண்டு வரலாம்.) சேர்க்க வேண்டிய படம் உள்ள தடத்தின் முகவரியை **(Path) File** பட்டையில் அச்சிட்டப் பிறகு **Open** என்ற பட்டியை 'கிளிக்' செய்யவும். தேர்ந்தெடுக்கப்பட்ட படம் இப்போது அட்டவணையில் தோன்றும். இவ்வாறு ஆப்பிள் படம் ஒன்றை தேர்ந்தெடுத்து அச்சிடப்பட்ட அட்டவணைத் தாளை படம் 6.26 இல் காணலாம்.



படம் 6.26 படத்துடன் உள்ள அட்டவணைத்தாள்

சிறப்புக் குறியீடுகளை அட்டவணையில் சேர்க்க **Insert Special Characters** என்ற பணிக்குறியை ‘கிளிக்’ செய். அப்பொழுது தோன்றும் **Special Characters** உரையாடல் பெட்டியில் இருந்து தேவையான சிறப்புக் குறியீட்டை தேர்வு செய்து **OK** என்ற பொத்தானை ‘கிளிக்’ செய்யவும் β என்ற சிறப்புக் குறியீட்டை இவ்வாறு தேர்வு செய்து சேர்க்கப்பட்ட அட்டவணைத் தாளை 6.27 இல் காணலாம்.



படம் 6.27 சிறப்புக் குறியீட்டு அச்சிடப்பெற்ற அட்டவணைத்தாள்

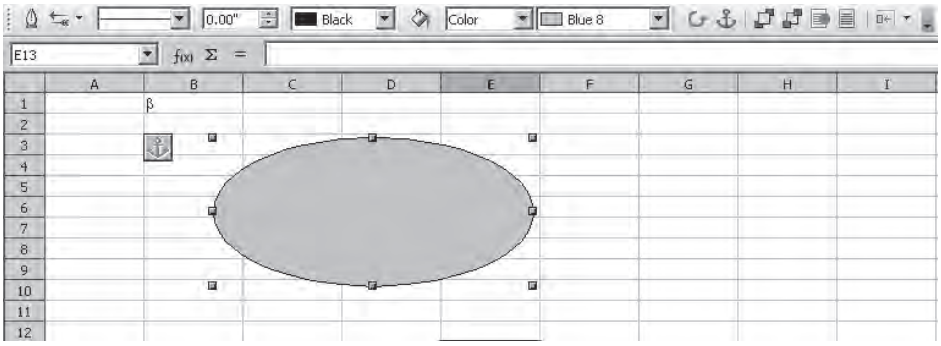
6.14 அட்டவணைத் தாளில் வரைபடம் வரைதல்

ஒரு அட்டவணைத்தாளுக்குள், கோடுகள், வட்டங்கள், நீள்வட்டங்கள் போன்றவற்றை வரைய **StarCalc**–இல் வசதிகள் உள்ளன. அதனைச் செய்ய முதன்மை கருவிப்பட்டையில் **Show Draw Functions** உள்ள என்னும் பணிக்குறியை ‘கிளிக்’ செய்யவும். உடனே படம் 6.28இல் கண்டவாறு **Draw Functions** பட்டை தோன்றும். அதிலிருந்து தேவையான எந்தக் கருவிக்குறியை **(Tool)** வேண்டுமானாலும் தேர்ந்தெடுத்துக் கொள்ளலாம். இவற்றின் உதவியால் வரைபடங்கள் வரையும் முறை கீழே தரப்பட்டுள்ளது.

ஒரு கருவிக்குறியை தேர்வு செய்து அதன் மீது ‘கிளிக்’ செய். எடுத்துக்காட்டாக, நீள்வட்டம் **(Ellipse)** வரைய கீழ்க்கண்டவாறு செய்க.

- **Ellipse** என்ற கருவிக்குறியை ‘கிளிக்’ செய்து, பிறகு சுட்டியை அட்டவணைத்தாளில் எந்த இடத்தில் நீள்வட்டம் வரைய விரும்புகிறோமோ அந்த இடத்தில் வைத்து சுட்டெலியை அழுத்தி இழுத்து நீள்வட்டத்தை வேண்டிய அளவிற்கு வரைந்துக்கொள்ளவும். இப்படி உருவாக்கிய நீள்வட்டம் படம் 6.29 காட்டப்பட்டுள்ளது.

படம் 6.28 Draw functions கருவிப்பட்டை

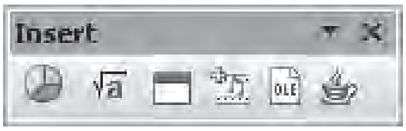


படம் 6.29 நீள்வட்டம் வரையப்பட்ட அட்டவணைத்தாள்

6.15 பொருள் (Object) சேர்த்தல் : (Inserting Object)

வரைபடங்கள், உருவப்படங்கள், வாய்ப்பாடுகள் போன்ற பொருள்களை அட்டவணைத்தாளில் சேர்க்க **StarCalc** வழி செய்கிறது. இதனைச் செய்ய **Insert Object** என்ற பணிக்குறியை **View →Toolbar** பணிக்குறிகளிலிருந்து 'கிளிக்' செய்யவும். உடனே ஒரு தற்காலிக கருவிப் பட்டை கீழ்க்கண்ட பணிக்குறிகளுடன் தோன்றும்.

1. **Insert Chart** பணிக்குறி
2. **Insert Formula** பணிக்குறி
3. **Insert Floating Frame** பணிக்குறி
4. **Insert Movie and Sound** பணிக்குறி
5. **Insert OLE Objects** பணிக்குறி
6. **Insert Applet** பணிக்குறி



இவற்றில் சில பணிக்குறிகள் மிகவும் உயர்நிலை பணிகளுக்கு பயன்படுவதால், அவை இந்த புத்தகத்தின் குறிக்கோளிற்கு அப்பாற்பட்டவை என்று கருதி விடப்படுகின்றன.

- **Insert Chart பணிக்குறி**

  இந்தப் பணிக்குறி அட்டவணைத்தாளில் உள்ள தரவுகளை பட்டை வரைபடங்களாகவோ **(Bar Charts),** அல்லது வட்ட வரைபடங்களாகவோ **(Pie - Charts),** கோட்டு வரைபடங்களாகவோ **(Line Charts),** அல்லது பிற வரைபட வடிவங்களாகவோ வரைந்துக் காட்டப் பயன்படுகின்றது. இது பற்றி விவரமாக பின்வரும் பத்திகளில் கொடுக்கப்பட்டுள்ளது.

- **Insert Formula பணிக்குறி**

  இந்தப் பணிக்குறி கணக்கிடப் பயன்படும் வாய்பாடுகளை அட்டவணைத் தாளில் சேர்க்கப் பயன்படுகிறது.

- **Insert Floating Frame பணிக்குறி**

  இந்தப் பணிக்குறிஅட்டவணைத்தாளிற்குள்உருள்திரை **(Rolling Screen)** உருவாக்கப் பயன்படுகிறது.

- **Insert Movie and Sound பணிக்குறி**

  இந்தப் பணிக்குறி ஒளிக்காட்சி, ஒலிக்கோப்புகளை அட்டவணையில் சேர்க்கப் பயன்படுகிறது.

- **Insert OLE Objects பணிக்குறி**

  இந்தப் பணிக்குறி பிற மென்பொருள்களில் பயன்பாட்டில் இருக்கும் பருப் பொருள்களை அட்டவணைத் தாளிற்குக் கொண்டுவர பயன்படுகிறது.

- **Insert Applet பணிக்குறி**

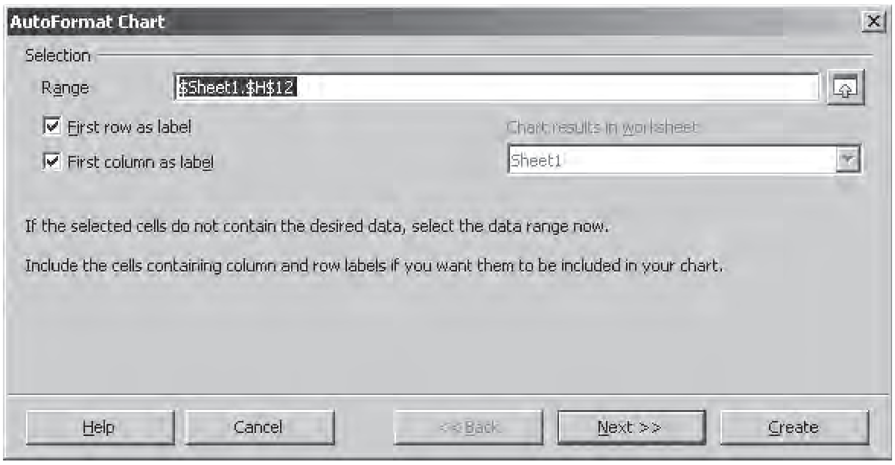
  இந்த பணிக்குறி **Applet** களை அட்டவணைத் தாளிற்குக் கொண்டுவரப் பயன்படுகிறது.

6.16 அட்டவணைத் தாளில் வரைபடங்களுடன் செயல்படல்

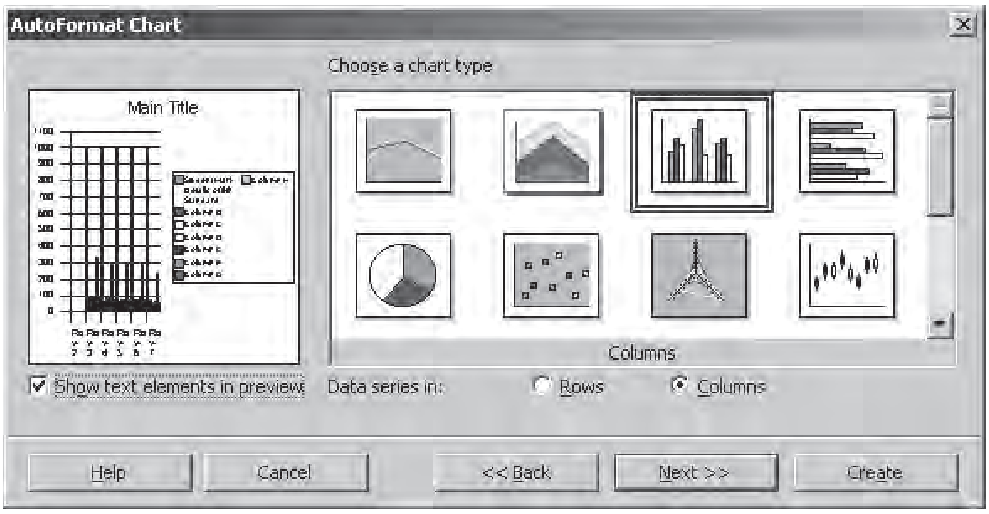
StarCalc – மென் பொருளில் உள்ள, பலராலும் பாராட்டப்பெற்ற ஒரு வசதி, அட்டவணைத்தாளில் உள்ள தரவுகளை வரைபடங்களாக வரைந்து காட்ட முடியும் என்பதாகும். இந்த வரைபடங்களின் நோக்கம், தரவுகளை பார்த்து, எளிதாகப் புரிந்துகொள்கின்ற வகையில் வரைபடங்களாக அளிக்க வேண்டும் என்பதே. தரவுகளை வரைபட வடிவில் கொடுக்க கீழ்க்கண்ட செய்முறையை பின்பற்றுக.

- வரைபடம் எந்த தரவிற்கு வரைய வேண்டுமோ அந்த தரவுகளை தேர்ந்தெடுத்துக் கொள்க.
- பிறகு **Insert → Chart** என்ற பட்டியை அல்லது Insert Chart அல்லது என்ற பணிக்குறியை 'கிளிக்' செய்யவும்.
- சுட்டி+ குறியீட்டுடன் கூடிய ஒரு சிறு வரைபடமாக தோன்றுவதைக் காணலாம். இந்த சுட்டியை அட்டவணைத்தாளில் எந்த இடத்தில் வரைபடத்தை சேர்க்க வேண்டுமோ அந்த இடத்திற்குக் கொண்டு சென்று 'கிளிக்' செய்யவும். இப்பொழுது **Auto Format Chart** சன்னல் திரை, படம் **6.30** இல் உள்ளவாறு தோன்றுவதைக் காணலாம்.
- அட்டவணைத்தாளில் எந்த பரப்பில் வரைபடத்தை வரைய வேண்டும் என்ற விவரத்தைக் கொடுக்குமாறு சன்னல்திரை நம்மைத் தூண்டும். எடுத்துக்காட்டாக, வரைபடப் பரப்பைக் குறிக்க **C3 : F7** என்று கொடுத்து **Next** பொத்தானை 'கிளிக்' செய்யவும். படம் **6.31** இல் கண்டவாறு வரைபடங்களின் வகைகள் முன்மாதிரி **(Preview)** படக்காட்சியுடன் ஒரு சன்னல் திரையில் தோன்றுவதைப் பார்க்கலாம். இப்பொழுது எந்த வகை வரைபடத்தில் தரவுகளைக் கொடுக்க விரும்புகின்றீர்களோ. அந்த வரைபடத்தை தேர்வு செய்யவும். உடனே இதன் முன்மாதிரிப் படம், சன்னல் திரையில் தோன்றுவதைப் பார்க்கலாம். பிறகு **Next** என்ற பொத்தானை 'கிளிக்' செய்யவும்.
- ஒவ்வொரு வகையான வரைபடத்திற்கும் வெவ்வேறு வடிவமைப்புகள் உள்ளன. இந்த வடிவமைப்புகள் படம் **6.32** இல் காட்டி இருப்பதைப் போல சன்னல் திரையில் தெரியும். அவற்றின் முன்மாதிரி படக்காட்சியும் உடனே தெரியும். அவற்றுள் எந்த வடிவமைப்பு தேவையோ அதனை தேர்வு செய்து கொள்ளலாம். தேர்வு செய்தபின் **Next** பொத்தானை 'கிளிக்' செய்யவும்.
- அடுத்த சன்னல்திரைப் படம் **6.33**–இல் உள்ளதைப் போல தோன்றும். இதில் வரைபடத்திற்கு தலைப்பு, **X** அச்சு, **Y** அச்சுகளுக்கான தலைப்புகள் ஆகியவற்றைக் கொடுக்க வேண்டும்.

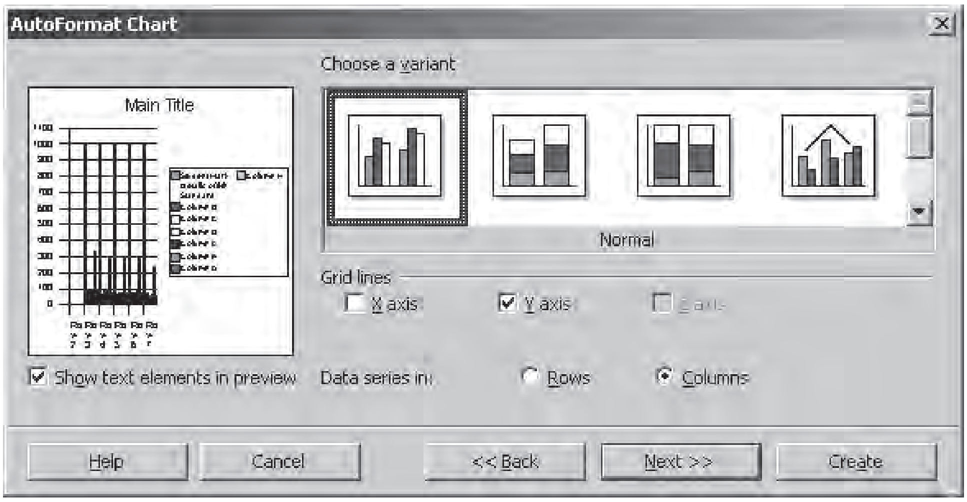
- இப்போது **Create** பொத்தானை 'கிளிக்' செய்யவும். உடனே அட்டவணைத்தாள் நீங்கள் விரும்பிய வரைபடத்துடன் படம் **6.34** இல் காட்டியுள்ளபடி தோன்றும்.



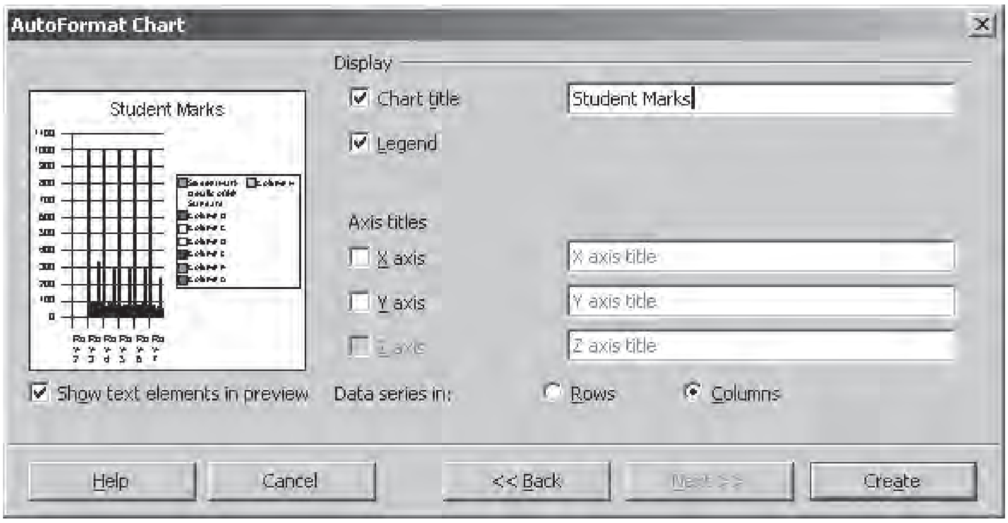
படம் 6.30 Auto Format சன்னல்திரை



படம் 6.31 StarCalc – இல் உள்ள வரைபட வகைகள்



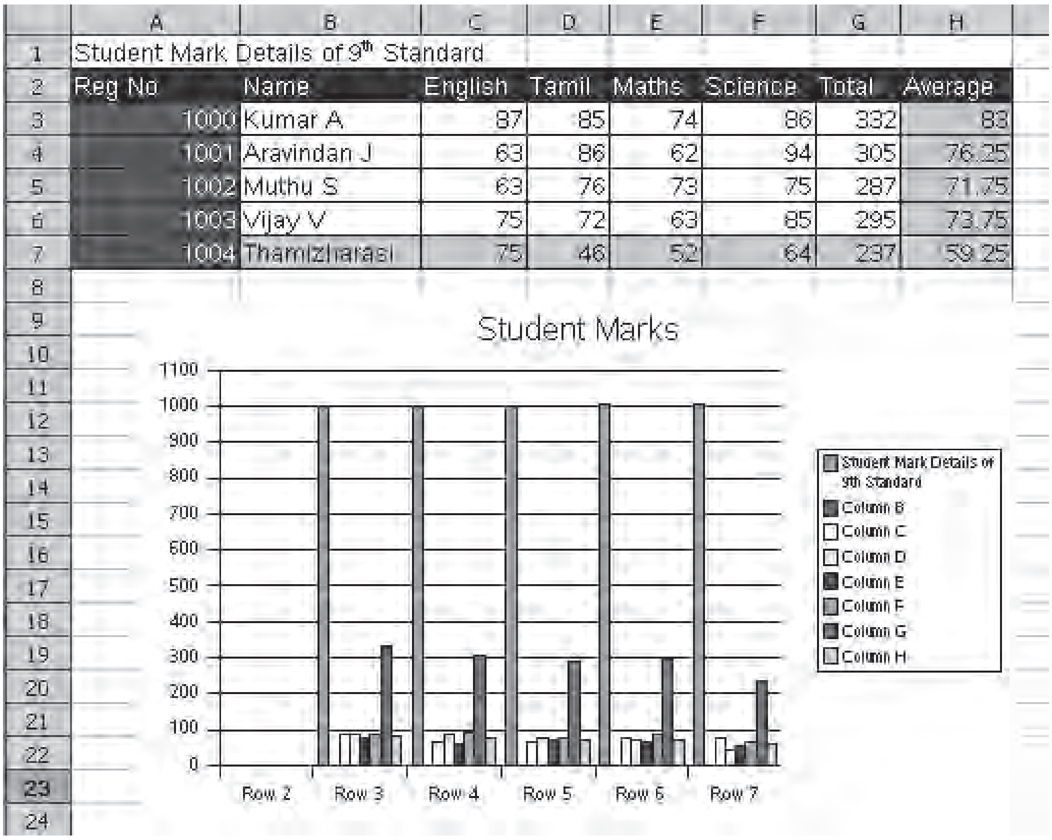
படம் 6.32 வரைபட வகைகளின் வெவ்வேறு வடிவமைப்புகள்.



படம் 6.33 தலைப்புகள், குறியீடுகள் கொடுக்க உதவும் திரை

செய்து பார்

மதிப்பெண் அட்டவணைத் தாளில் உள்ள மாணவர்களின் மதிப்பெண்களின் வேறுபாட்டை காட்ட ஒரு கோட்டு வரைபடம் வரைக.



Student Mark Details of 9th Standard

Reg No	Name	English	Tamil	Maths	Science	Total	Average
1000	Kumar A	87	85	74	86	332	83
1001	Aravindan J	63	86	62	94	305	76.25
1002	Muthu S	63	76	73	75	287	71.75
1003	Vijay V	75	72	63	85	295	73.75
1004	Thamizharasi	75	46	52	64	237	59.25

படம் 6.34 வரைபடத்தைக் காட்டும் அட்டவணைத்தாள்.

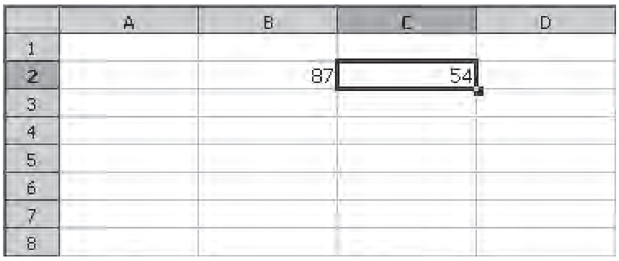
6.17 ஒன்றுக்கு மேற்பட்ட அட்டவணைக் கோப்புகளில் செயலாற்றல்

StarCalc –இல் ஒரு அட்டவணைக் கோப்பு பல தாள்களைக் கொண்டதாக இருக்கலாம். ஒவ்வொரு அட்டவணைத் தாளுக்கும் அதற்கென ஒரு பெயர் உண்டு. அட்டவணைத்தாளின் அடிப்பகுதியில் தாள்களின் பட்டியல், பொத்தான்களைப் போல கொடுக்கப்பட்டிருக்கும். ஒரு குறிப்பிட்ட தாளைத் தேர்வு செய்ய அந்தத் தாளின் பெயர் உள்ள பொத்தானை ‘கிளிக்’ செய்ய வேண்டும். உடனே அந்தத் தாள் வெள்ளை நிறத்தில் தோற்றம் அளிக்கும்.

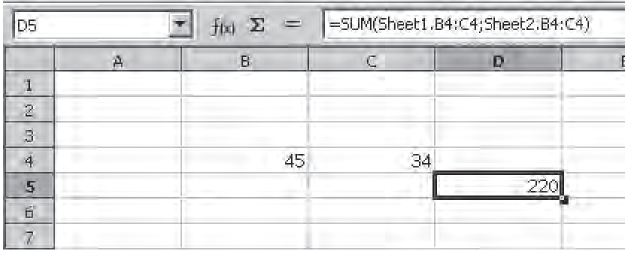
ஒவ்வொரு அட்டவணைத் தாளையும் தனித்தனியே பயன்படுத்தலாம். ஒரு தாளில் உள்ள தரவுகளை இன்னொரு தாளில் பார்க்கவோ அல்லது கணக்கீடுகளில் பயன்படுத்தவோ செய்யலாம். எடுத்துக்காட்டாக, தரவுகளை தாள் 1, தாள் 2 இரு தாள்களில் எழுதி, அந்தத் தரவுகளைப் பயன்படுத்தி தாள் 2 இல் கணக்கீடுகளைச் செய்யலாம். அதாவது தாள் ஒன்றின் A1 நுண்ணறையில் உள்ள எண்ணை தாள் 2இல் **A1** நுண்ணறையில் உள்ள எண்ணுடன் கூட்டுத் தொகையை தாள் 2–இன் **A3** நுண்ணறையில் எழுதி சேமிக்கலாம். இதனைச் செய்ய **= SUM (Sheet1.A1 : Sheet2.A1)** என்ற முப்பரிமாண வாய்பாட்டை (3D) **Formula** கொடுக்க வேண்டும்.

இன்னொரு எடுத்துக்காட்டாக தாள் ஒன்றின் **B4, C4** நுண்ணறைகளில் எண்கள் 87, 54 முறையே இருப்பதாகக் கொள்வோம். (படம் 6.35). தாள் இரண்டின் **B4, C4** நுண்ணறைகளின் எண்கள் 45, 34 முறையே இருப்பதாகக் கொள்வோம். இப்பொழுது இரண்டு தாள்களிலும் உள்ள **B4, C4** நுண்ணறைகளில் தேக்கப்பட்டுள்ள எண்களை கூட்டி தாள் இரண்டின் **D5** நுண்ணறையில் எழுதுவதாகக் கொள்வோம். இதனைச் செய்ய.

= SUM (Sheet1.B4 : C4 ; Sheet2.B4 : C4) என்ற முப்பரிமாண வாய்பாட்டை (**3D Formula**) கொடுக்கவும். இந்த வாய்பாட்டில் **B4** க்கும் **C4** க்கும் இடையே உள்ள : என்ற குறி ஒவ்வொரு தாளிலும் உள்ள **B4, C4** நுண்ணறைகளின்மதிப்புகளைத் தனித்தனியே கூட்ட வேண்டும் என்பதையும், இந்த வாய்பாட்டிற்குள் இருக்கும்; என்ற குறி இரண்டு தாள்களின் கூடுதல்களையும் கூட்ட வேண்டும் என்பதையும் குறிக்கின்றன. இந்த கணக்கீடுகளை படங்கள் 6.35, 6.36 காட்டுகின்றன.



படம் 6.35 தாள் 1– இல் தரவுகளுடன் உள்ள அட்டவணைத்தாள்

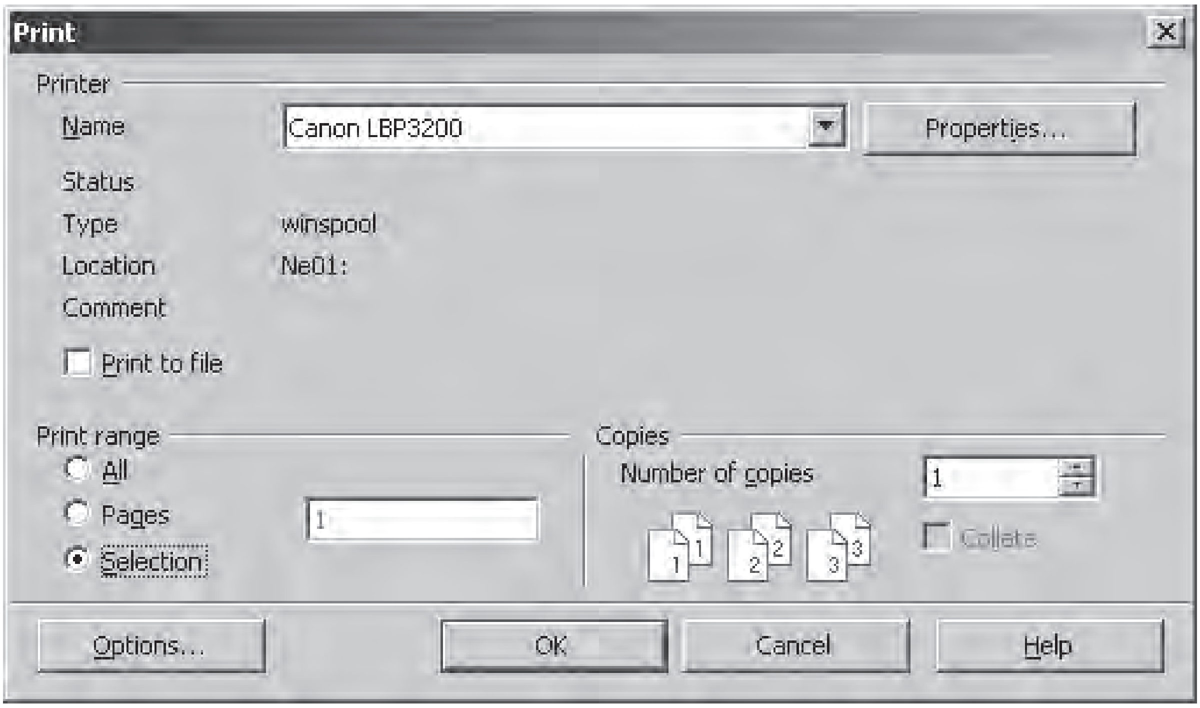


படம் 6. 36 தாள் 2 – இல் தரவுகளும் விடைகளும் உள்ள அட்டவணைத்தாள்

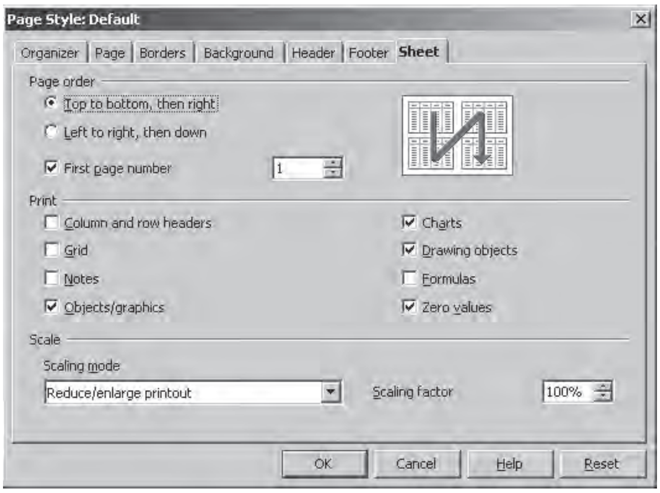
6.18 அட்டவணைத் தாள்களை அச்சிடல்

சார்புப் பட்டையில் உள்ள **(Function Menu)** அச்சிடல் **(Print)** என்னும் பணிக்குறியை 'கிளிக்' செய்தால், நாம் உருவாக்கிய கோப்பில் உள்ள எல்லா தாள்களிலும் இருக்கின்ற தரவுகள் அனைத்தும் அச்சிட்டுக் கொடுக்கப்படும். அட்டவணைத்தாளில் உள்ள எந்த பகுதியை வேண்டுமானாலும் நான் விரும்பியவாறு அச்சிட்டுக் கொள்ளலாம். அப்படிச் செய்ய, அச்சிடப்பட வேண்டிய பகுதியைத் தேர்வு செய்து **Format→Print Range → Edit** என்ற பணிக்குறியை 'கிளிக்' செய்யவும். அதன் பிறகு **Print** பணிக்குறியை 'கிளிக்' செய்தால் தேர்வு செய்த பகுதி மட்டுமே அச்சிடப்படும். அச்சிடத் தேர்வு செய்த பகுதியை நீக்க வேண்டும் எனில் **Format→ Print Range →Remove** என்ற பணிக்குறிகளை 'கிளிக்' செய்யவும்.

File பட்டியில் உள்ள **Print** பட்டியை பயன்படுத்தியும் அட்டவணைத் தாள்களை அச்சிடலாம். **File,Print** பட்டிகளைக் 'கிளிக்' செய்தவுடன் அச்சு உரையாடல் பெட்டி **(Print Dialogue Box),** படம் 6.37 இல் காட்டியுள்ளவாறு தோன்றும். அச்சு உரையாடல் பெட்டியில் **Print Range** இன் கீழ் **All** என்ற பொத்தானை தேர்வு செய்தால் கோப்பின் எல்லாத் தாள்களும் அச்சிடப்படும். **Pages** என்ற பொத்தானை தேர்வு செய்து அதன் கீழ்வரும் **Selection** என்ற பொத்தானை தேர்வு செய்தால் தேர்வு செய்யப்பட்ட பகுதிகள் மட்டும் அச்சிடப்படும். File பட்டியில் உள்ள **Page Preview** என்ற பட்டியைப் பயன்படுத்திஅச்சிடப்பட வேண்டிய அட்டவணைத் தாளை முன்கூட்டியே பார்த்துக் கொள்ளலாம். மேற்கூறிய வழிமுறையை பயன்படுத்தினால் அட்டவணைக் கட்டங்கள் இன்றி அச்சிடப்படும். அட்டவணைக் கட்டங்களுடன் அச்சிட வேண்டும் எனில் **Format** பட்டியில் உள்ள **Page** என்ற பட்டியைத் தேர்வு செய்து, வருகின்ற உரையாடல் பெட்டியில் உள்ள **Sheet** என்னும் பொத்தானை 'கிளிக்' செய்யவும். பிறகு **Grid** என்ற தேர்வுப் பெட்டியை **(Check Box)** 'கிளிக்' செய்தால் அட்டவணைத் தாள் கட்டங்களுடன் அச்சிடப்படும்.



படம் 6.37 Print உரையாடல் பெட்டி



படம் 6.38 Page Styles உரையாடல் பெட்டி

6.19 StarCalc –இன் தரவுத்தள சார்புகள் (Database Functions)

தரவுத் தளங்களைப் போலவே **StarOffice** இன் அட்டவணைச் செயலியைக் கொண்டும் தரவுத்தளங்களை இயக்க முடியும். அதாவது தரவுத்தளங்களின் தரவுகளைத் தேடுதல், வகைப்படுத்தல் போன்ற மற்றும் பல கணிப்பீடுகளை அட்டவணைச் செயலியில் உள்ள தரவுத்தள சார்புகளைக் கொண்டு செயலாற்ற முடியும். தரவுத் தளங்கள் பற்றி விரிவாக அடுத்த அதிகாரத்தில் படிப்போம்.

சுருக்கம்

- அட்டவணைச் செயலி செயலாற்ற உதவும் ஒரு கருவி.
- ஒரு அட்டவணைச் செயலி மென்பொருள், எண்களைக் கொண்டு கணக்கிடவும், கணக்கீட்டு ஆய்வுகளைச் செய்யவும் பயன்படுகிறது.
- ஒரு அட்டவணைத் தாள் நேர் வரிசைகளையும் வரிசைகளையும் கொண்டுள்ளது. வரிசைகளும் நேர்வரிசைகளும் குறுக்கிடும் போது உண்டாகும் கட்டத்திற்கு நுண்ணறை என்று பெயர். ஒரு நுண்ணறை அந்த நுண்ணறையை உருவாக்கும் நெடுவரிசைப் பெயர், வரிசை எண் ஆகியவற்றால் அறியப்படுகின்றது. ஒரு நுண்ணறையில் எண்கள், உரைகள், வாய்பாடுகள் இவற்றைக் கொடுத்து சேமிக்கலாம்.
- அட்டவணைச் செயலியைப் பயன்படுத்தி உருவாக்கிய ஒரு தரவுக் கோப்பை அட்டவணைத் தாள் என்று அழைக்கின்றோம்.
- அட்டவணைத் தாளை உருவாக்கல்.
- ஒரு அட்டவணைத்தாளை உருவாக்கும் பணியில் தரவுகளைச் சேர்த்தல், தரவுகளை உள்ளிடல், வாய்பாடுகளை உருவாக்கல், பதிப்பாய்வு செய்தல், வடிவமைத்தல், வரைபடங்களைச் சேர்த்தல், தரவுகளை ஆய்தல், அட்டவணைத்தாளை அச்சிடல் போன்ற செயல்பாடுகள் அடங்கியுள்ளன.
- ஒரு வாய்பாடு நுண்ணறைகளில் உள்ளத் தரவுகளைப் பயன்படுத்தி வாய்பாட்டில் கண்டவாறு கணக்கிட்டு விடையை ஒரு நுண்ணறையில் எழுதச் செய்கிறது.

- நுண்ணறைகளில் உள்ளத் தரவுகளைப் பயன்படுத்தி வாய்பாட்டின் படி கணக்கிட்டு விடையை ஒரு நுண்ணறையில் எழுதச் செய்கிறது.
- வாய்பாடுகளில் உள்ள எண் கணித இயக்கங்கள் கணித இயலில் செயற்படுத்தப்படும். அதே வரிசையில், அதாவது, வர்க்கம் காணல், எண்களை எதிர்மறையாக்கல், பெருக்கல்/வகுத்தல், கூட்டுதல், கழித்தல் என்ற வரிசை முறையில் அட்டவணைச் செயலியிலும் செயல்படுத்தப்படுகின்றன.
- சார்புகள் அட்டவணைச் செயலியில் முன்னதாகவே உருவாக்கப்பட்ட வாய்பாடுகளாகும். StarCalc –இல் பலவகையான சார்புகள் உள்ளன.
- நுண்ணறைப் பார்வையிடல் (Cell Referencing) செயற்பாட்டை ஒப்பீட்டு நுண்ணறை முகவரி முறையையோ அல்லது தனித்த நுண்ணறை முகவரி முறையையோ பயன்படுத்தி செயல்படுத்தலாம்.
- அட்டவணைத்தாளில் உள்ள தரவுகளை தடித்த எழுத்துக்களாகவோ, சாய்ந்த எழுத்துக்களாகவோ, அல்லது சொற்களை கோடிட்டோ வடிவமைத்துக் காட்டலாம். தரவுகளை வரிசைப்படுத்துதல், தரவின் எழுத்துக்களையும் / எண்களையும் வெவ்வேறு வடிவுகளிலும், அளவுகளிலும், வண்ணங்களிலும் மாற்றியமைத்தல் போன்ற செயற்பாடுகளையும் செய்யலாம். மேலும், எண்களை பணக்குறியீடுகளுடனும், விழுக்காடுகளாலும், தசம எண்களுடனும் காட்டலாம்.
- அட்டவணைத்தாளில் படங்கள், சிறப்புக் குறியீடுகள், நாம் வரைந்த ஓவியங்கள் போன்றவற்றைச் சேர்க்கலாம்.
- அட்டவணைத்தாளில் தரவுகளை எளிதாகப் புரிந்துக் கொள்ள அவற்றை வரைபடங்களாக வரைந்துக் காட்டலாம்.
- அட்டவணைச் செயலி, வகைப்படுத்தல், தேடுதல், அச்சிடல் போன்ற தரவுத்தள அமைப்புகளின் ஆட்சித் திறன்களையும் பெற்றதுள்ளது.

பயிற்சி

I. கோடிட்ட இடத்தை நிரப்புக.

1. கணக்கீட்டுகள் செய்யவும் கணக்கீட்டு ஆய்வுகள் செய்யவும் ________________ பயன்படுத்தலாம்.
2. எண்களை வரைந்து காட்டும் முறைக்கு ______________________ என்று பெயர்.
3. அட்டவணைச் செயலியைக் கொண்டு உருவாக்கும் கோப்பிற்கு ____________ என்று பெயர்.
4. எண் அல்லாத தரவுகளுக்கு _________________ என்று பெயர்.
5. _________________________இது இயக்க A நுண்ணறைகளை (Active Cells) குறிக்கின்றது.
6. அட்டவணைச் செயலியில் வாய்பாட்டை குறிக்க __________________ குறியீடு முதலில் பயன்படுத்தப்படுகிறது.

7. கணிப்பீடுகளை தானே செயல்படுத்த ____________________ பயன்படுகிறது.

8. ________________, ______________________ என்பவை இருவகையான நுண்ணறை முகவரி முறைகளாகும்.

9. எண் தொடர்களை உருவாக்க _____________________ கட்டளை பயன்படுகிறது

10. ஒரு நுண்ணறையின் நெடுவரிசை, வரிசை ஆகியவற்றின் தலைப்புகள் நுண்ணறையின் ___________________ என்று அழைக்கப்படுகின்றன.

11. பிற பயன்பாடுகளிலிருந்து குறிப்புகளை அட்டவணையில் சேர்க்க ____________________ பயன்படுகிறது.

12. முதன்மைப் பட்டியலிலுள்ள __________________________ என்ற வசதி அட்டவணையை அச்சிடும் முன் பார்க்கப் பயன்படுகிறது.

II. கீழ்க்கண்ட வினாக்களுக்கு விடையளி

1. அட்டவணைச் செயலியை சுருக்கமாக வரையறுத்து அதன் அடிப்படை அமைப்பை விளக்குக.

2. எண்களைக் கொண்டு கணக்கிடும் திறனாற்றலைத் தவிர அட்டவணைச் செயலியில் உள்ள பிற வசதிகளை வரிசைப்படுத்தி விளக்குக.

3. ஒரு நுண்ணறையில் வாய்பாட்டைக் கொடுக்கும் பொழுது எண்களையும் இயக்கக் குறியீடுகளையும் ஒரு குறிப்பிட்ட வரிசையில் தான் கொடுக்க வேண்டுமா ? உன் விடைக்கு விளக்கம் தருக.

4. ஒரு நுண்ணறையில் உள்ள தரவை மாற்ற முடியுமா ? ஆம் எனில் எப்படி ?

5. அட்டவணைச் செயலியைக் கொண்டு ஒரு எண் வரிசையை எப்படி உருவாக்குவாய் ? எடுத்துக்காட்டுடன் விளக்குக.

6. சார்புகள் என்றால் என்ன ? அவற்றை அட்டவணைத்தாளில் எப்படிப் பயன்படுத்துவாய் ? எடுத்துக்காட்டுடன் விளக்குக.

7. தேதி கணக்கீடு என்றால் என்ன ? விளக்குக.

8. தரவுகளை நுண்ணறைகளுக்குக் கொடுக்கும் போது பணக்குறியீடுகள் கால்புள்ளி, விழுக்காட்டுக் குறியீடு போன்றவற்றை கொடுக்கத் தேவையில்லை. ஏன் என்று சுருக்கமாக விளக்குக.

9. அட்டவணைச் செயலியின் எந்த செயற்கூறு, தரவுகளை எளிதாகப் பார்த்துப் புரிந்துகொள்ளும் வகையில் கொடுக்க உதவுகிறது ?

10. பல அட்டவணைத்தாள்களுடன் செயல்படல், அட்டவணைகளை அச்சிடல் ஆகியன பற்றி விளக்குக.

அதிகாரம் 7

தரவுத்தளம்

7.1 முன்னுரை

நாம் கணிப்பொறிகளை பல்வேறு வகைகளில் பயன்படுத்துகிறோம். பயன்பாடுகளை அறிவியல் ஆய்வுப் பயன்பாடு அல்லது தரவுச் செயலாக்கப் பயன்பாடுகள் எனப் பிரிக்கலாம். ஊதியப் பட்டியல், தேர்வு முடிவு ஆய்வுகள், வங்கிக் கணக்கீடுகள், தொடர்வண்டி/ பேருந்து பயணச்சீட்டு முற்பதிவுகள், வரவு – செலவு திட்டமிடல், பொருள்பட்டியல் கட்டுப்பாடு போன்ற பயன்பாடுகளை கணிப்பொறியில் செயற்படுத்தும் முறையை தரவுச் செயலாக்கம் **(Data Processing)** என்று கூறுகின்றோம்.

இந்த அதிகாரத்தில் தரவுச் செயலாக்கம் பற்றியும், தரவுச் செயலாக்க மென்பொருள், குறிப்பாக Star Base பற்றியும் படிப்போம்.

7.2 தரவும் தகவலும்

7.2.1 தரவு

தரவைக் குறிக்கும் **Data** என்ற சொல் தகவல் கூறு என்ற பொருள் கொண்ட **Datum** என்ற சொல்லிலிருந்து வருகின்றது. **Datum** என்ற சொல் ஒருமையையும், **Data** என்ற சொல் பன்மையையும் குறிக்கும்.) தரவு **(Data)** என்பது மக்கள், இடங்கள், பொருள்கள் ஆகியவற்றை பற்றிய தகவல் கூறுகளைக் குறிப்பதாகும் . கணிப்பொறியைப் பொறுத்தமட்டில் தரவு என்பது கணிப்பொறியின் மென்பொருள்களில் பயன்படுத்தப்படும் மாறிகளின் மதிப்பைக் குறிக்கும்.

எடுத்துக்காட்டாக

கீழ்க்கண்ட எடுத்துக்காட்டிலுள்ள தொடர்களை நோக்கவும்.

விஜயனின் வயது 16.

விஜயன் 12ஆம் வகுப்பில் படிக்கிறான்.

விஜயன் கணிதத் தேர்வில் 80% மதிப்பெண் பெற்றுள்ளான்.

மேலே கூறப்பட்ட தகவல்கள் விஜயன் என்பவரைச் சேர்ந்ததாகும். இந்தத் தகவலை கணிப்பொறியில் கொடுத்து செயலாக்க வேண்டுமெனில், அதற்கான கட்டளைக் கோப்பில், பெயர், வயது, வகுப்பு, பாடம். மதிப்பெண் என்ற ஐந்து மாறிகளை (Variables) வரையறுக்க வேண்டும். பிறகு இந்த மாறிகளுக்கு மேற்கண்ட சொற்தொடர்களிலிருந்து மதிப்புகளை எடுத்துக் கீழே கண்டவாறு கொடுக்க வேண்டும்;

பெயர்	=	விஜயன்
வகுப்பு	=	12
வயது	=	16
மதிப்பெண்	=	80
பாடம்	=	கணிதம்

மேலே கூறப்பட்ட எடுத்துக்காட்டில் ஐந்து மாறிகளுக்கும் கொடுக்கப்பட்ட ‘விஜயன்’, 12, 16,80 ‘கணிதம்’ ஆகியவை தரவுகள் எனப்படும்.

கீழே கொடுத்துள்ள இன்னொரு எடுத்துக்காட்டை பார்க்கவும்.

சென்னையிலிருந்து கோயமுத்தூர் 500 கல் தொலைவில் உள்ளது.

கோயமுத்தூர் கடல் மட்டத்திலிருந்து 1500 அடி உயரத்தில் உள்ளது. இந்த இரு தொடர்களும் கோயமுத்தூர் பற்றிய இரு தகவல்களை கொடுக்கின்றன. இந்த எடுத்துக்காட்டில் 'நகரம்', 'தொலைவு' , 'இடம்' , 'உயரம்' என்ற நான்கு மாறிகளின் மதிப்புகளாகிய ' கோயமுத்தூர்' , '500', 'சென்னை' , '1500' ஆகியவை தரவுகளாகும்.

தரவுகள் பல வகைப்படும். தரவு வகைகளை **(Data types) எழுத்து (Character) வகை, எண் (Numeric) வகை, பூலியன் (Boolean) வகை** எனப் பிரிக்கலாம். எண் தரவு வகையை முழு எண்கள் **(Integer)** தசமப்புள்ளி எண்கள் **(Floating Point Numbers)** என்றும், இரட்டிப்புத் துல்லிய **(Double Precision)** எண்கள், தேதி, நேரம் என்று, பல வகைகளாகப் பிரிக்கலாம். கணிப்பொறி மொழிகளும் **(Programing Languages)** மென்பொருள் தொகுப்புகளும் **(Software Packages)** முழு எண், தசமப்புள்ளி எண் போன்ற எண் வகைகளை முன்னதாகவே உருவாக்கப்பட்டு தன்னுள் கொண்டுள்ளன. இப்படிப்பட்ட தரவு வகைகள் தொடக்க நிலைத் தரவு வகைகள் **(Primitive data types)** எனக் கூறப்படுகின்றன. ஆனால், சரக்கு வழங்குவோரின் எண்கள் **(Supplier's Number),** பொருள்களின் உறுப்புப்பெயர்கள், நகரப்பெயர்கள், நிறங்கள் போன்ற பல பயன்பாடுகளுக்குத் தொடக்க நிலைத் தரவு **(Primitive data type)** வகைகள் மட்டுமே போதாது. இப்படிப்பட்ட, பயன்பாடுகளுக்கு பயனர்கள் **(Users)** தரவுகளுக்கு வேண்டிய பொருத்தமான தரவு வகைகளை வரையறுத்துக் கொள்ள தரவுத்தள அமைப்புகள் **(Data base systems)** வழி செய்கின்றன். இப்படி பயனர்கள் தாங்களே வரையறுத்துக் கொள்ளும் தரவு வகைகளுக்கு ' பயனாளர்' தரவு வகை **(User Defined data type)** என்று பெயர்.

7.2.2 தகவல் (Information)

இனி நாம் கணிப்பொறியின் பார்வையில் தகவல் **(Information)** என்றால் என்ன என்பதைக் காண்போம். கொடுக்கப்பட்ட தரவுகளுக்கு இடையே நிலவும் உறவை வெளிப்படுத்துமாறு செயற்படுத்தப்பட்ட தரவுகளின் தொகுப்பையே தகவல் **(Information)** என்பர். தரவுகளைச் செயற்படுத்துவது என்றால் என்ன ? பல மாறிகளின் **(Variable)** தரவுகளை கணிப்பீடுகளுக்கு **(Computations)** அல்லது இயக்கங்களுக்கு **(Operations)** ஆட்படுத்தி, அதன்மூலம் ஒவ்வொரு மாறியையும் பிற மாறியுடன் உறவுபடுத்தி, தரவுகளைப் பொருள்படச் செய்வதையே தரவுத் செயலாக்கம் **(Date Processing)** என்கிறோம். இவ்வாறு, ஏதேனும் பொருளை உணர்த்தும் உறவுபடுத்தப்பட்ட தரவுகளின் தொகுப்பை தகவல் **(Information)** என்பர். மேற்கண்ட எடுத்துக்காட்டில், 'பெயர்' என்ற மாறியின் மதிப்பாகிய ' விஜயன்' என்ற தரவையும் 'வயது' என்ற மாறியின் மதிப்பாகிய '16' வயது என்ற தரவையும் தொடர்புபடுத்தி,

விஜயனின் வயது 16

என்ற தகவலைப் பெறலாம். அதே எடுத்துக்காட்டில், 'பெயர்' 'வகுப்பு' என்ற மாறிகளின் தரவுகளாகிய 'விஜயன்' '12' என்ற தரவுகளைத் தொடர்புபடுத்தும் போது

விஜயன் படிப்பது 12 ஆம் வகுப்பு

என்ற தகவலைப் பெறுகின்றோம். அதே போல, 'கோயமுத்தூர்' , 'சென்னை' , '500' என்ற தரவுகளைத் தொடர்பு படுத்தப்படும் போது

‘கோயமுத்தூர் சென்னையிலிருந்து **500** கல் தொலைவில் உள்ளது.’ என்ற தகவல் கிடைக்கின்றது.

7.3 தரவுச் செயலாக்கம் (Data Processing)

ஆசிரியர் தயாரிக்கும் மதிப்பெண் பட்டியலை எடுத்துக் கொள்வோம்’ இந்தப் பட்டியலில், மாணவர்களின்’ பதிவெண், ‘பெயர்’ பாடங்களில் பெற்ற மதிப்பெண்கள், தேர்ச்சி/வீழ்ச்சி என்ற நிலை போன்ற விவரங்கள் இருக்கும். இந்த விவரங்களை ‘பதிவெண்’, ‘பெயர்’, ‘தமிழ்’, ‘ஆங்கிலம்’, ‘கணிதம்’, ‘இயற்பியல்’, ‘வேதியியல்’, மேலும் ‘முடிவு’ என்ற மாறிகளின் தரவுகளைக் கொடுத்து அட்டவணை 7.1–இல் காட்டியுள்ளதைப் போல மதிப்பெண் பட்டியல் தயாரிக்கப்படுகின்றது.

அட்டவணை 7.1 : 11 – ஆம் வகுப்பு ‘அ’ பிரிவின் மதிப்பெண் பட்டியல்

Mark list of XI STD – A Section

Reg. No.	Name	Tamil	Eng.	Maths	Phy.	Chem	Comp.	Pass/ Fail
1001	Anbu A	80	85	99	95	82	88	P
1002	Aruna S	85	90	90	92	90	98	P
1003	Balu S	35	56	95	75	70	80	F
1004	Xavier D.	80	99	95	96	97	99	P
1005	Banu M	75	80	56	50	60	70	P
1006	Chandran M	40	32	58	45	26	70	F
1007	Abdul Kader M	80	80	90	88	80	95	P

இந்த மதிப்பெண் பட்டியலைத் தயாரிக்க, வகுப்பாசிரியர் முதலில் ஒவ்வொரு பாட ஆசிரியரிடமிருந்தும் மாணவர்களின் பதிவெண், பெயர், அவரது பாடத்தில் மாணவர்கள் பெற்ற மதிப்பெண்கள் போன்றவற்றைக் கேட்டுப் பெறுகின்றார். அத்துடன் தலைமை ஆசிரியரிடமிருந்து தேர்ச்சி பெறுவதற்கான விதிமுறைகள் பற்றிய விவரங்களையும் பெறுகின்றார். பிறகு,எல்லா ஆசிரியர்களிடமிருந்து பெற்ற மதிப்பெண் விவரங்களை அட்டவணை 7.1 இல் காட்டியுள்ளவாறு தொகுத்து பட்டியலிடுகின்றார். தொகுத்தவுடன், தொகுத்த விவரங்களில் ஏதேனும் பிழை ஏற்பட்டுள்ளதா என முதலில் தானே சரிபார்கின்றார். பிறகு ஒவ்வொரு பாட ஆசிரியரிடமும் பட்டியலைக் கொடுத்து அவரவர் பாட மதிப்பெண்கள் பிழையின்றி எழுதப்பட்டுள்ளனவா என சரி பார்த்துப் பெறுகின்றார். இவ்வாறு மதிப்பெண்கள் பிழையின்றி கொடுக்கப்பட்டுள்ளனவா என்று உறுதி செய்து கொண்டபின் தேர்ச்சி, விதிமுறைகளைப் பயன்படுத்தி மாணவர்களின் தேர்ச்சி அல்லது வீழ்ச்சி நிலையை முடிவு செய்கின்றார். பிறகு, இந்த முடிவுகளை மதிப்பெண் பட்டியலில் எழுதி சரிபார்க்கின்றார். சரிபார்த்த முடிவுகளைக் கொண்ட மதிப்பெண் பட்டியலை தலைமை ஆசிரியரின் ஒப்புதலுக்கு அனுப்புகின்றனார். தலைமை ஆசிரியரின் ஒப்புதல் அளித்த பின் அட்டவணை 7.2 –இல் காட்டியுள்ளவாறு, மாணவர் பதிவெண், பெயர், தேர்வின் முடிவுகள் (தேர்ச்சி/ வீழ்ச்சி) போன்ற விவரங்களைக் கொண்ட பட்டியலைத் தயாரித்து அறிவிப்புப் பலகையில் போடுகின்றனர்.

அட்டவணை 7.2 : 11–ஆம் வகுப்பு மாணவர்களின் தேர்ச்சி அறிக்கை

Reg. No.	Name of the Student	Result
1001	Anbu A	Pass
1002	Aruna S	Pass
1003	Balu S	Fail
1004	Xavier D.	Pass
1005	Banu M	Pass
1006	Chandran M	Fail
1007	Abdul Kader M.	Pass

நாம் இதுவரை விவரித்த செயற்பாடுகள் மாணவர்களின் தேர்வறிக்கை செயற்பாடாகும். அதாவது மாணவர்களின் பெயர்கள், பதிவெண்கள், மதிப்பெண்கள் போன்ற தரவுகளை செயற்படுத்தி தேர்வு முடிவுகள், அறியப்பட்டன. எனவே இப்படிப்பட்ட செயற்பாட்டை தரவுச் செயலாக்கம் **(Data Porcessing)** என்கின்றோம். தரவுச் செயற்பாடு எந்த வகைப் பயன்பாட்டை வேண்டுமானலும் சார்ந்து இருக்கலாம்.

மேற்கண்ட எடுத்துக்காட்டில் விவரித்ததைப் போல தரவுச் செயலாக்கம், தரவுகளைச் சேகரித்தல் **(Data Collection),** சரிபார்த்தல் **(Verification),** செல்லுபடியாக்குதல் **(Validation),** அறிக்கை தயாரித்தல் போன்ற பல பணிக்கூறுகளைக் கொண்டுள்ளது. கணிப்பொறிகள் பயன்பாட்டிற்கு வருமுன் தரவுச் செயலாக்கங்கள் கையாலேயே செய்யப்பட்டு வந்தன. இப்போது கூட பல நிறுவனங்கள் தரவுச் செயலாக்கங்களை கையாலேயே செய்து வருகின்றன.

மேற்கண்ட எடுத்துக்காட்டில் மாணவர்களின் தேர்வறிக்கைச் செயலாக்கம், தொடக்கத்திலிருந்து, அதாவது மதிப்பெண்களை ஒவ்வொரு பாட ஆசிரியரிடமிருந்தும் சேகரித்ததிலிருந்து, இறுதியில் தேர்வு அறிக்கை தயாரித்தது வரை எல்லா செயற்பாடுகளையும் வகுப்பாசிரியர் தன் கைப்பட தானே செயல்பட்டு முடித்துள்ளார். இப்படிக் கையாலேயே தரவுகளைச் செயற்படுத்தும் முறைக்குக் கைவழித் தரவுச் செயலாக்கம் **(Manual Data Processing)** என்று பெயர் கையால் செயல்படுவதற்குப் பதிலாக கணிப்பொறியைப் பயன்படுத்தி.

- மதிப்பெண் பட்டியல் தயாரித்தல்
- தரவுகளை தேக்கிப் பாதுகாத்தல்
- தேக்கியத் தரவுகளைப் பயன்படுத்தி முடிவுகளை ஆய்ந்து அறிதல்
- முடிவுகளை விரும்பிய வடிவமைப்பில் அச்சிட்டுப் பெறுதல்.

போன்ற செயற்கூறுகளை செய்து முடிப்போமேயானால், இப்படிப்பட்ட செயற்பாடுகளுக்கு ‘கணிப்பொறி வழித்தரவுச் செயலாக்கம்’ **(Computerised Data Processing)** என்று பெயர்.

7.3.1 கைவழிச் செயலாக்க முறையில் ஏற்படும் சில குறைபாடுகள்.

கைவழித்தரவுச் செயலாக்க முறையில் உள்ள சில குறைபாடுகள் கீழே கொடுக்கப்பட்டுள்ளன :

1. கைவழிச் செயலாக்கப் பணியை மேற்கொள்ளும் போது செய்கையின் திறனும் துல்லியமும் இதனைச் செய்யும் தனி மனிதரின் திறமையைப் பொறுத்தே அமைகின்றது.

2. கைவழிச் செயலாக்கப் பணியை மேற்கொண்டால் பணியை முடிக்க மிகுந்த நேரம் ஆகும்.

3. பொதுவாக மனிதர்கள், அவர்களாக செயலபடும் போது கணிப்பீட்டுப் பிழைகளும் **(Computational)** நோக்கு மயக்கப்பிழைகளும் **(Parallax Errors)** இழைப்பது இயல்பு.

4. நாமே இப்பணியில் கைவழிச் செயலாக்க முறையில் ஈடுபடும் போது ஒவ்வொரு கட்டத்திலும் தாள்களை மிகுதியாக பயன்படுத்த வேண்டியிருக்கும். இதனால் நாம் பாதுகாப்பாக வைத்திருக்க வேண்டிய தாள்களும் கோப்புகளும் பெருகி, அவை ஒரு பெரும் குவியலாகிவிடும்.

5. தரவுகளைத் திருத்துதல், மாற்றுதல், நீக்குதல் போன்ற செயற்பாடுகள் கைவழித்தரவுச் செயலாக்கத்தில் பெரும் சோர்வையும் தளர்ச்சியையும் தரும்.

7.3.2 கணிப்பொறி வழித்தரவுச் செயலாக்கத்தின் நன்மைகள்.

கணிப்பொறி வழிச் செயலாக்க முறையில் ஏற்படும் சில நன்மைகள் கீழே கொடுக்கப்பட்டுள்ளன :

1. தகவல் கூறுகளைச் சேகரித்து கணிப்பொறியில் ஒரு முறை கொடுத்து முடித்து விட்டால் பிறகு தரவுகளைத் திருத்துதல்,மாற்றுதல், நீக்குதல் போன்ற பல செயற்பாடுகளை மிகுந்த அலுப்பு இன்றி செய்து முடிக்கலாம். இதனால் நம் உழைப்பும் நேரமும் வீணாவது இல்லை.

2. கணிப்பொறி வழி செயலாக்கத்திற்குத் தேவையான கட்டளைக் கோப்பை **(Program)** உருவாக்கவும். அவற்றைப் பிழையின்றி சோதித்துப் பின், பயன்பாட்டிற்கு ஏற்பாடு செய்யவும் சில காலம் ஆகலாம். ஆனாலும் ஒரு தடவை ஒரு கட்டளைக் கோப்பை உருவாக்கி பயன்படுத்தத் தொடங்கிவிட்டால் பின் அதன் மூலம் மிக வேகமாக எவ்வளவு தகவல் கூறுகளையும் எத்தனை முறை வேண்டுமானாலும் மிக எளிதாகக் கொடுத்து செயல்படுத்தலாம்.

3. கணிப்பொறி வழியாகத் தரவுகளை செயல்படுத்தும் போது தவறுகள் ஏற்படும் வாய்ப்புகள் மிகக் குறைவு.

4. கணிப்பொறியில் ஏராளமான தரவுகளையும், தகவல்களையும் மிகச் சிறிய தேக்கப்பகுதியில் தேக்கி வைக்க முடியும். எனவே ஏராளமான தாட்கள், கோப்புகள், பதிவேடுகள் இவற்றை பெருங்குவியலாகச் சேர்த்து வைக்க வேண்டியத் தேவையில்லை.

5. தற்போது கணிப்பொறி வலையமைப்புகள் மிகவும் பெருகி வருகின்றன. இம்முறையில் ஒரு கணிப்பொறியிலிருந்து இன்னொரு கணிப்பொறிக்கு, தகவல்களை வெகு வேகமாகவும் எளிதாகவும் அனுப்பவோ பெறவோ முடியும். எடுத்துக்காட்டாக, தொடர்வண்டி பயணச்சீட்டு பதிவு இப்போதெல்லாம் கணிப்பொறி வலையமைப்பைப் பயன்படுத்தி செய்யப்படுகின்றது. இந்த வலையமைப்பில் உள்ள ஒரு மையக் கணிப்பொறியில், தொடர்வண்டிகள் மற்றும் பயணிகள் பதிவு பற்றிய விவரங்கள், இவற்றை செயலாக்கத் தேவையான கட்டளைக் கோப்புகள் போன்றவை தேக்கி வைக்கப்படுகின்றன. இந்த மையக் கணிப்பொறி சென்னையில் இருப்பதாகக் கொள்வோம். பிற நகரங்களில் (எ.கா. திருச்சியில் உள்ள ஒரு பதிவெழுத்தர் தனது கணிப்பொறி முனையத்திலிருந்து மையக் கணிப்பொறியுடன் தொடர்பு கொண்டு, சென்னை கணிப்பொறியில் உள்ள விவரங்களை அறிந்து, அவர் திருச்சியில் உள்ள ஒரு பயணிக்கு சென்னையிலிருந்து திருச்சி செல்வதற்கான பயணச்சீட்டை பதிவு செய்ய முடியும்.)

6. கணிப்பொறித் தரவுத்தளங்களில், தகவல்களைத் தேடுதல், வகைப்படுத்தல், கோப்புகளைச் சேர்த்தல் போன்ற பல தரவுச் செயற்பாடுகளை மிகவும் திறம்படச் செய்ய முடிகின்றது.

7.4 தரவுத்தளம் (Data base)

ஒரு தரவுத்தளம் என்பது தொடர்புடைய தரவுகள் அல்லது தகவல்களின் தொகுப்பைக் கொண்ட ஒரு களஞ்சியமாகும். தரவுத்தளத்தில் தரவுகள் ஒரு குறிப்பிட்ட அமைப்பில் ஒழுங்குபடுத்தப்பட்டு இருக்கும். எடுத்துக்காட்டாக, அட்டவணை 7.3–இல் ஒரு கல்வி நிறுவனத்தின் பணியாளர் தரவுத்தளம் **(Staff database)** தொடர்பான விவரங்கள் கொடுக்கப்பட்டுள்ளன. இதில் பணியாளர்களின் பணி எண் , பெயர், தகுதி, பதவி, துறை, பணியில் சேர்ந்த நாள், மாத ஊதியம் போன்ற தகவல்கள் ஒரு குறிப்பிட்ட அமைப்பில் கொடுக்கப்பட்டுள்ளன. இந்த தரவுத்தளத்தில் ஏழு புலங்கள் **(files)** உள்ளன. இரண்டு புலங்களின் **(1&7)** தரவுவகை எண் வகைத்தரவாகவும், நான்கு புலங்கள் **(2,3,4,&5)** எழுத்துவகைத் தரவாகவும், ஒன்று தேதிவகைத் தரவாகவும் அமைக்கப்பட்டுள்ளன. ஒரு தரவுத்தளத்தில் எத்தனை பதிவுகள் **(Records)** வேண்டுமானலும் இருக்கலாம்.

நிறுவன பணியாளர் தரவுத்தளத்தில் (அட்டவணை 7.3)

1001 Arul A BE (CSE) Lecturer Com. Sc. & Engg. 23/01/96 10000 என்பது ஒரு பதிவு **(Record)** ஆகும்.

மேற்கண்ட நிறுவனத் தரவுத் தளத்தைப்போல் வெவ்வேறு தரவுத்தளங்களை அமைக்கலாம். எடுத்துக்காட்டாக அட்டவணை 7.4 –இல் கொடுக்கப்பட்டுள்ளது.

அட்டவணை 7.3 : நிறுவன பணியாளர் தரவுத்தளம்

Emp. No.	Staff Name	Qualification	Designation	Department	Date of Joining	Monthly Salary (Rs.)
5001	Arul A.	B.E (CSE)	Lecturer	Comp. Sc. & Engg	23/01/96	10000
5002	Aruna B.	M.E. (CSE)	Sr. Lecturer	-do-	18/06/98	15500
5003	Ezhil M.	M.Tech.	Sr. Lecturer	Elec.Comm. Engg.	10/05/99	15000
5004	Thiagu N	M.B.A	Lecturer	Dept of Management	19/03/96	10000
5005	Laura I.	M.E. (CSE)	Sr. Lecturer	Comp.Sc. & Engg	01/07/99	15000

அட்டவணை **7.4 :** கணிப்பொறித்துறைப் பணியாளர் தரவுத்தளம்

Emp. No.	Staff Name	Qualification	Designation	Department	Date of Joining	Monthly Salary (Rs.)
5001	Arul A.	B.E (CSE)	Lecturer	Comp. Sc. & Engg.	27/01/97	10000
5002	Aruna B.	M.E. (CSE)	Sr. Lecturer	-do-	18/06/98	15500
5005	Laura I.	M.E. (CSE)	Sr. Lecturer	Comp. Sc. & Engg.	01/07/99	15000

ஒரு கல்வி நிறுவனத்தின் கணிப்பொறித்துறை பற்றிய தரவுத்தளமாகும். இதுபோல் உங்கள் நண்பர்கள், உறவினர்கள் ஆகியோரின் முகவரிகளைக் கொண்ட ஒரு தரவுத்தளத்தை நீங்கள் அமைத்துக் கொள்ளலாம்.

ஒரு தரவுத்தளம் பெரும்பாலும் ஒன்று அல்லது ஒன்றிற்கும் மேற்பட்ட வரிசைகளையும் நெடுவரிசைகளையும் கொண்ட இருபரிமாண **(two dimensional)** அட்டவணைகளைக் கொண்டதாக இருக்கும். ஒன்றோடொன்று தொடர்புடைய தரவுகளின் தொகுப்பைக் கொண்ட ஒரு அட்டவணையை ஒரு கோப்பு **(File)** என்கிறோம். ஒரு அட்டவணையின் ஒவ்வொரு வரிசையையும் ஒரு பதிவு **(Record)** எனப்படும். முன்பே குறிப்பிட்டதைப் போல ஒரு பதிவு என்பது ஒரு தரவுத்தளத்தில் கொடுக்கப்படும் ஒவ்வொரு உருப்படி **(item/entity/entry)**யின், தொடர்புடைய தரவுகளின் தொகுப்பாகும். அட்டவணையின் ஒவ்வொரு நெடுவரிசையும் ஒரு புலம் **(field)** எனப்படும். ஒவ்வொரு புலமும் ஒரு தரவு வகையைச் **(Data type)** சேர்ந்ததாக இருக்கும்.

7.4.1 தரவுத்தளத்தைக் கையாளுகை (Manipulation of a Database)

தரவுத்தளத்தைக் கீழ்க்கண்ட ஒன்று அல்லது மேற்பட்ட வழிகளில் கையாளலாம்:

- தேடல் **(Searching)**
- வரிசையாக்கம் **(Sorting)**
- சேர்த்தல் **(Merging)**
- தரவுகளைக் கொண்டு கணிப்பீடு செய்தல்**(performing calculations on data)**
- வடிகட்டல் **(Filtering)**
- தரவுத்தளத்தை பதிப்பாய்வு செய்தல் **(Editing the database)**
- அறிக்கை உருவாக்கல் **(Report Generation)**
- **தேடல் (Searching)**

  ஒரு தரவுத்தளத்திலிருந்து தேவைப்பட்ட ஒரு குறிப்பிட்ட தரவை தேடிக் கண்டுபிடித்து எடுக்கும் செயற்பாட்டிற்குத் தேடல் **(Searching)** என்று பெயர். எடுத்துக்காட்டாக, மொத்த மதிப்பெண்களில் முதலாவதாக வந்த மாணவன் யார் என்று அறியவோ, அல்லது ஒவ்வொரு பாடத்திலும் முதல் மதிப்பெண் எடுத்த மாணவன் யார் என்று அறியவோ, அல்லது ஒரு குறிப்பிட்ட மாணவன் எல்லா பாடங்களிலும் எடுத்த மதிப்பெண்கள் பற்றி அறியவோ விரும்பினால் தரவுத்தள கட்டளைகளைப் பயன்படுத்தி தேடிப் பெறலாம்.

- **வரிசையாக்கம் (Sorting)**

ஒரு அட்டவணையில் உள்ள தரவுகளை ஒரு குறிப்பிட்ட வரிசையில் ஒழுங்குபடுத்தி எழுதுவதற்கு வரிசையாக்கம் **(Sorting)** என்று பெயர். மேற்கண்ட நிறுவன தரவுத்தளத்தில் (அட்டவணை 7.3), நாம் விரும்பினால் பணியாளர் பட்டியலை அகர வரிசைப்படுத்தி எழுதலாம் அல்லது பதவியின் தகுதிப்படி வரிசைப்படுத்தி எழுதலாம் அல்லது பதவியில் சேர்ந்த தேதிப்படி வரிசைப்படுத்தி எழுதலாம்.

மாணவர் தரவுத்தளத்திலிருந்து தேர்ச்சி பெற்ற மாணவர்கள் பட்டியலையும் தேர்ச்சி பெறாத மாணவர்களின் பட்டியலையும் தனித்தனியே வரிசைப்படுத்தி ஏறு வரிசையிலோ அல்லது இறங்கு வரிசையிலோ எழுதலாம்.

- **சேர்த்தல் (Merging)**

ஒரே அல்லது வெவ்வேறு தரவுத்தளங்களின், ஒன்று அல்லது ஒன்றுக்கு மேற்பட்ட அட்டவணைகளிலிருந்து தரவுகளை ஒன்று சேர்க்கும் செயற்பாட்டை சேர்த்தல் **(Merging)** என்று அழைக்கின்றோம்.

- **தரவுகளைக் கொண்டு கணிப்பீடு செய்தல் (Performing Calculations on data)**

தரவுத்தளத்தில் சேமித்து வைக்கப்பட்டுள்ள தரவுகளைக் கொண்டு நாம் எந்த வகையான எண் கணித **(arithmetic)** கணக்கீடு **(Calculations)** களையும் செய்யலாம். எடுத்துக்காட்டாக, மாணவர் தரவுத்தளத்தில், ஒரு மாணவன் எல்லா பாடங்களிலும் எடுத்துள்ள மதிப்பெண்களைக் கூட்டி அவன் எடுத்துள்ள மொத்த மதிப்பெண்ணைக் கணக்கிட்டு தனியொரு புலத்தில் **(field)** எழுதலாம்.

- **வடிகட்டல் (Filtering)**

சில நேரங்களில் ஒரு அட்டவணை முழுவதையும் கணிப்பொறித் திரையில் பார்க்க இயலாத அளவிற்கு அட்டவணை பெரிதாக இருக்கலாம். அப்படிப்பட்ட நேரங்களில் வடிகட்டல் **(Filtering)** முறையைக் கையாண்டு அட்டவணையில் உள்ள எந்தத் தகவல்களை மட்டும் பார்க்க விரும்புகின்றோமோ அந்தத் தகவல்களை மட்டும் திரையில் தோன்றச் செய்து பார்க்கலாம்.

தரவுத்தளங்களில், ஒரு அட்டவணையிலிருந்து தேர்வு செய்யப்பட்ட ஒரு குறிப்பிட்ட பட்டியல் அல்லது ஒரு பதிவின் சிறு பகுதியை மட்டும் பார்க்கவோ அல்லது மேலோடித் **(Browsing)** தேடவோ, உருவாக்கப்பட்டுள்ள செயற்கூறே வடிகட்டலாகும் **(Filtering),** பயனர்கள் கொடுக்கும் நிபந்தனைகளை **(Conditions)** நிறைவு செய்யும் பதிவுகள் **(Record)**மட்டுமே திரையில் தெரியும். நிபந்தனைகளை நிறைவு செய்யாத பதிவுகள் மறைத்து வைக்கப்படும்.

- **தரவுத்தளத்தை பதிப்பாய்வு செய்தல் (Editing the database)**

ஒரு தரவுத்தளத்தில் ஏற்கனவே உள்ள தரவுகளைத் திருத்துதல், ஏற்கனவே உள்ள தரவுகளை **(Data),** தரவுப் புலங்களை **(Fields),** அல்லது தரவுப் பதிவுகளை **(Records)** நீக்குதல், புதிய தரவுகளை, புலங்களை அல்லது பதிவுகளைத் தரவுத்தளத்தில் சேர்த்தல், அல்லது தரவுத்தளத்தின் வடிவமைப்பை மாற்றி அமைத்தல் அல்லது மேலும் இது போன்ற செயற்பாடுகளைச் செய்தல், ஆகிய செயற்பாடுகளை பதிப்பாய்வு **(Editing)** செய்தல் என்று கூறுகின்றோம்.

- **அறிக்கைகள் உருவாக்கல் (Report Generation)**

  அறிக்கை என்பது தேவையான தரவுகளை தரவுத்தளத்திலிருந்து பெற்று உருவாக்கும் ஒரு ஆவணம். தரவுத்தளத்தின் தரவுகளிலிருந்து நமக்குத் தேவையான அறிக்கைகளை உருவாக்கிக் கொள்ளலாம். எடுத்துக்காட்டாக, மாணவர் தரவுத்தளத்திலிருந்து தேர்ச்சி பெறாத மாணவர்களின் அறிக்கை ஒன்றும், தேர்ச்சி பெற்ற மாணவர்களின் அறிக்கை ஒன்றும், இவை போன்ற பிற அறிக்கைகள் பலவும் உருவாக்கலாம்.

7.4.2 தரவுத்தள வகைகள் (Database Types)

தரவுத்தள அமைப்பின் தத்துவ அடிப்படையைக் கொண்டு, தரவுத் தளங்களை கீழ்க்கண்டவாறு பல வகைகளாகப் பிரிக்கலாம்:

1. ஒற்றைக் கோப்புத் தரவுத்தளம் **(Flat- file Data base)**
2. உறவு நிலைத் தரவுத்தளம் **(Relational Data base)**
3. படிநிலைத் தரவுத்தளம் **(Hierarchical Data base)**
4. வலையமைப்பு தரவுத்தளம் **(Network Data base)**
5. பொருள் நோக்குத் தரவுத்தளம் **(Object - Oriented Data base)**

- **ஒற்றைக் கோப்புத் தரவுத்தளம் (Flat- file Data base):**

ஒரே ஒரு அட்டவணையைக் கொண்டு அமைக்கப்பட்ட தரவுத்தளத்திற்கு ஒற்றைக் கோப்பு **(Flat-file)** தரவுத்தளம் என்று பெயர். முகவரிகள், பொருட்பட்டியல் போன்ற ஒருவர் பயன்படுத்தும் சிறிய அளவு தரவுத்தளங்களை அமைக்க ஒற்றைக் கோப்புத் தரவுத்தளங்கள் பயனுள்ளதாக இருக்கும். அட்டவணைச் செயலி, ஒற்றைக் கோப்புத் தரவுத்தளத்திற்கு ஒரு எடுத்துக்காட்டாகும். ஒற்றைக் கோப்புத் தரவுத்தளங்கள், அடிப்படையில் எளிமையாக இருந்தாலும், அவை பல் அட்டவணை **(Multifile table)** தரவுத்தளங்களோடு ஒப்பிடும் போது குறிப்பிடத்தக்க குறைபாடுகளைக் கொண்டுள்ளன. அவை சிக்கலான வினவல்களை **(Complex requests)** ஒப்புக் கொள்வதில்லை. பல் அட்டவணைச் செயலாக்கத்திற்கு இவை ஏற்றவை அல்ல.

- **உறவுநிலைத் தரவுத்தளம் (Relational Data base):**

தொடர்புள்ள அட்டவணைகளைக் கொண்டு உறவுநிலை அமைப்பின் அடிப்படையில் அமைக்கப்படும் தரவுத்தளங்களை உறவுநிலைத் **(Relational)** தரவுத்தளங்கள் என்கிறோம். ஒரு உறவுநிலைத் தரவுத்தளத்தில், இரண்டு அல்லது அதற்கு மேற்பட்ட அட்டவணைகளில், உள்ள ஒன்று அல்லது பல புலங்கள் **(Fields)** பொதுவாக இருந்து, அட்டவணைகளுக்கு இடையே உறவை உண்டாக்குகின்றன. அந்த பொதுவான புலம் **(Fields)** அல்லது புலங்களுக்கு **(Field)** திறவுகோல்கள் **(Keys)** என்று பெயர்.

ஒரு முதன்மைத் திறவுகோல் **(Primary Key)** என்பது ஒரு தரவுத்தள அட்டவணையின் ஒரு பதிவை **(Record)** தனித்தன்மையோடு குறிக்கும் ஒரு திறவுகோலகும். உறவுநிலை தரவுத்தளங்களில் முதன்மை திறவுகோல் ஒன்று அல்லது பல புலங்களைக் கொண்டதாக இருக்கலாம். எடுத்துக்காட்டாக, அட்டவணை 7.3 –இல் பணியாளர் எண் **(Emp.No)** ஒரு முதன்மைத்திறவுகோல் பணியாளர் எண் ஒரு குறிப்பிட்ட பணியாளரின் பதிவை மட்டுமே

குறிப்பதால் இது முதன்மைத்திறவுகோல் எனப்படுகின்றது. உறவு நிலைத்தரவுத்தள அமைப்பு இன்றைய வணிகத்துறையில் மிகவும் பரவலாகப் பயன்பட்டு வரும் தரவுத்தளமாகும்.

- **படிநிலைத் தரவுத்தளம் (Hierarchical Data base):**

படிநிலைத் **(Hierarchical)** தரவுத்தளம் பெரும்பாலும் தலைமைக் கணிப்பொறி **(Mainframe Computer)** அமைத்தலில் பயன்படுத்தப்பட்டது. படிநிலை தரவுத்தளங்களில் பதிவுகள் **(record)** மரக்கிளைகளைப் போன்றதொரு அமைப்பில் **(tree - like structure)** ஒழுங்குபடுத்தப்படுகின்றன. பதிவுகளுக்கு இடையே உள்ள உறவு ஒரு குழந்தைக்கு ஒரு பெற்றோர் போன்ற பெற்றோர் –குழந்தை உறவைப் போல, கீழ்நிலையில் உள்ள ஒரு பதிவு மேல்நிலையில் உள்ள ஒரு பதிவுடன் மட்டுமே தொடர்புடையதாக இருக்கும்.

- **வலையமைப்பு தரவுத்தளம் (Network Data base):**

வலையமைப்புத் தரவுத்தளம் படிநிலைத் தரவுத்தளத்தைப் போலவே இருக்கும். ஒரே ஒரு வேறுபாடு என்னவெனில், வலையமைப்புத் தரவுத்தளத்தில் எந்த ஒரு பதிவும் **(record)** எத்தனை பதிவுகளுடன் வேண்டுமானலும் தொடர்பு கொண்டிருக்கலாம்.

- **பொருள் நோக்குத் தரவுத்தளம் (Object - Oriented Data base):**

பொருள் நோக்குத் தரவுத்தளம் ஒரு புதிய அமைப்பாகும். இது அண்மைக் காலத்திலிருந்து பெரும் ஆர்வத்தை உருவாக்கி வருகின்றது. இது தரவுகளைத் தரவுத்தளங்களில் கையாள மிகவும் வேறுபட்ட முறையைக் கொடுக்கின்றது. பொருள் நோக்கு **(Object - Oriented)** அமைப்பில், தரவுகளையும் அவற்றின் சிறப்பியல்புகள், பண்புகள், செயல்முறைகள் ஆகியவற்றையும் ஒன்று சேர்த்து பொருள் **(Object)** என்ற பெயரில் சிக்கலான ஒரு உருப்படியாக வரையறுக்கப்படுகின்றன. இயலபாக ஒரு பொருள் என்பது, ஒரு வினைபொருள், நிகழ்ச்சி, வீடு, பயன்பாட்டுக்கருவி, கலைப்பொருள், பயனாளர் புகார், கொள்முதல் பொருள் போன்ற எதுவாக வேண்டுமானலும் இருக்கலாம். ஒரு பொருள் அதன் சிறப்பியல்புகள் **(Characteristics)** பண்புகள் **(attributes)** செய்முறை **(Procedure)** ஆகியவற்றால் வரையறுக்கப்படுகின்றது. பொருளின் சிறப்பியல்புகள் **(Characterstics)** என்பன அமைப்பு **(Text)** ஒலி **(Sound),** வரைபடம் **(Graphics),** ஒளித்தோற்றம் **(Video),** போன்றவை என்று கூறலாம். பண்பு **(attribute)** களுக்கான எடுத்துக்காட்டுகளாக பொருளின் நிறம் **(Colour)** பருமன் **(Size),** பாணி **(Style),** அளவு **(Quantity),** விலை **(Price)** போன்றவற்றைக் கூறலாம். ஒரு பொருளைக் **(Object)** கையாளும் முறை அல்லது செயற்படுத்தும் முறையை செய்முறை **(Program)** என்று கூறலாம்.

பொருள் நோக்கு முறையை நன்றாகப் புரிந்துகொள்ள கீழ்க்கண்ட எடுத்துக்காட்டைப் பாருங்கள்:

கிளி, குருவி, காக்கை இவற்றிற்கு பொதுவான பண்புகள் உண்டு.

அவையாவன: இரண்டு இறக்கைகள், ஒரு அலகு, இரண்டு கால்கள், ஒருவால் இவற்றின் பொதுவான சிறப்பியல்பு இவை பறக்கவல்லன.

இப்படிப்பட்ட பொது இயல்புகளையும் பண்புகளையும் கொண்ட ஒரு இனத்தை பறவை என்கிறோம். பொருள்நோக்கு அமைப்பு முறையில் இதனை (பறவையை) ஒரு வகுப்பு **(Class)** என்பர். இந்தப் பொதுப் பண்புகள் ‘பறவை’ யென்ற வகுப்புடன் **(Class)** இணைப்பட்டு

அந்த வகுப்பின் பண்புகளாக வரையறுக்கப்படுகின்றன **(Encapsulation)** இப்பொழுது, அந்தையைப் பார்த்திராத ஒரு குழந்தையிடம் ஆந்தை ஒரு பறவை என்று கூறினால், அந்த குழந்தை உடனே, ஒகோ, ஆந்தை ஒரு பறவையென்றால் அதற்கு இரண்டு இறக்கைகள், ஒரு அலகு, இரண்டு கால்கள், ஒருவால் இருக்க வேண்டும் என்றும் அதனால் பறக்க முடியும் என்பதையும் குழந்தை தானே புரிந்துகொள்கின்றது அல்லவா? இதனை மரபுரிமைப்பெறல் **(Inheritance)** என்பர். கிளி, குருவி, காக்கை, ஆந்தை போன்றவை பறவை என்ற வகுப்பின் பொருள் **(Objects)** களாகும். இந்த பறவை என்ற வகுப்பின் பொருள்களாகிய **(Objects)** இந்த பறவைகள் ஒவ்வொன்றிற்கும் தனிப்பண்புகளும் உண்டு. அதாவது ஒவ்வொன்றின் அமைப்பு வேறு, பருமன் வேறு, நிறம் வேறு, அவை ஒவ்வொன்றும் எழுப்பும் ஒலி வேறு. இந்தத்தனிப் பண்புகள் அந்தந்த பறவைகளோடு சேர்த்து அறியப்படுகின்றன **(Encapsulated to each Object),** இப்படி உருவாக்கப்படுகின்ற அமைப்பு முறைக்குத்தான் பொருள் நோக்கு **(Object - Oriented)** அமைப்பு முறை என்று பெயர். இந்தத் தத்துவத்தைப் பயன்படுத்தி அமைக்கப்படும் தரவுத்தளத்திற்கு பொருள்நோக்குத் தரவுத்தளம் என்று பெயர்.

7.5 தரவுத்தள மேலாண்மையின் அடிப்படைத் தத்துவங்கள் Basic Concepts of Database Management System (DBMS)

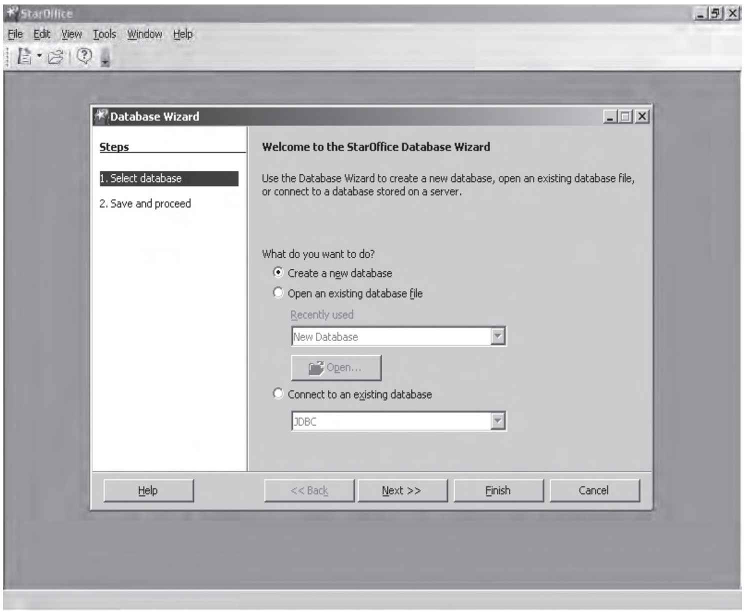
தரவுத்தள மேலாண்மை அமைப்பு **(DBMS)** என்பது, பல பயனர்கள், தரவுத்தளத்தில் உள்ள தரவுகளைப் பெறவோ, தேவையெனில் மாற்றவோ அல்லது எளிய சிக்கலான வினவல்கள் மூலம் பெறுவதற்கோ அல்லது செயற்படுத்துவதற்கோ தேவையான நிரல் **(Program)**களின் தொகுப்பாகும். தரவுத்தள மேலாண்மை அமைப்பின் **(DBMS)** மிகப்பெரும் வலிமையே பெரும் தரவுத் தளங்களிலிருந்து தரவுகளை மிகமிக விரைவில் பார்க்கவும் பெறவும் வழி செய்யும் வல்லமை கொண்டது என்பதாகும். ஒரு தரவுத்தள மேலாண்மை அமைப்பு திறன்மிக்க கணிப்பொறியில் செயல்படுத்தும்போது, எந்த ஒரு தரவையும் மிகப் பெரும் தரவுத் தளங்களிலிருந்தும் கூட சில மணித்துளிகளில், ஒரு நொடிக்கும் குறைவான நேரத்தில், கண்டு அறிய முடியும். தரவுத்தள மேலாண்மை அமைப்பின் செயற்பாடுகளை கீழ்க்கண்ட மூன்று பொதுப்பிரிவுகளாகப் பிரிக்கலாம்.

1. தரவுத்தளத்தில் தரவுகளை உள்ளிடல்
2. தரவுத்தளத்தின் பதிவுகளை திரும்ப ஒழுங்குபடுத்துதல் **(Reordering)**
3. தரவுகளின் பகுதிகளை தரவுத்தளத்திலிருந்து பெறல்

தரவுத்தள மேலாண்மை அமைப்பு, கணிப்பொறிகளின் வலையமைப்பு மூலம், பல பயனர்கள் ஒரே நேரத்தில் தரவுகளை பார்க்கவும் பங்கிட்டுக் கொள்ளவும் வழி செய்கின்றது.

7.6 Star Base–இல் செயலாற்றல்

Star Base–இல் செயலாற்ற **Star Office** என்ற மென்பொருளை முதலில் இயக்க வேண்டும். படம் 7.1 –இல் காட்டியவாறு **Star Office** –இன் தொடக்க Star Office –இன் சன்னல் திரை தோன்றும்.

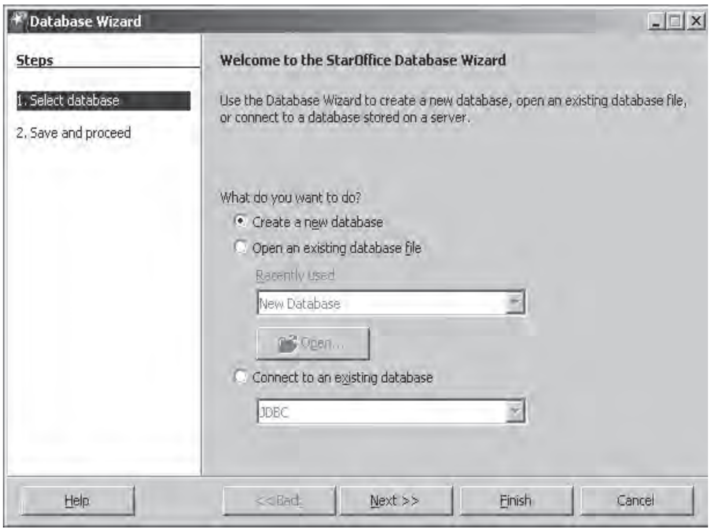


படம் 7.1 தொடக்க சன்னல் திரை

குறிப்பு:

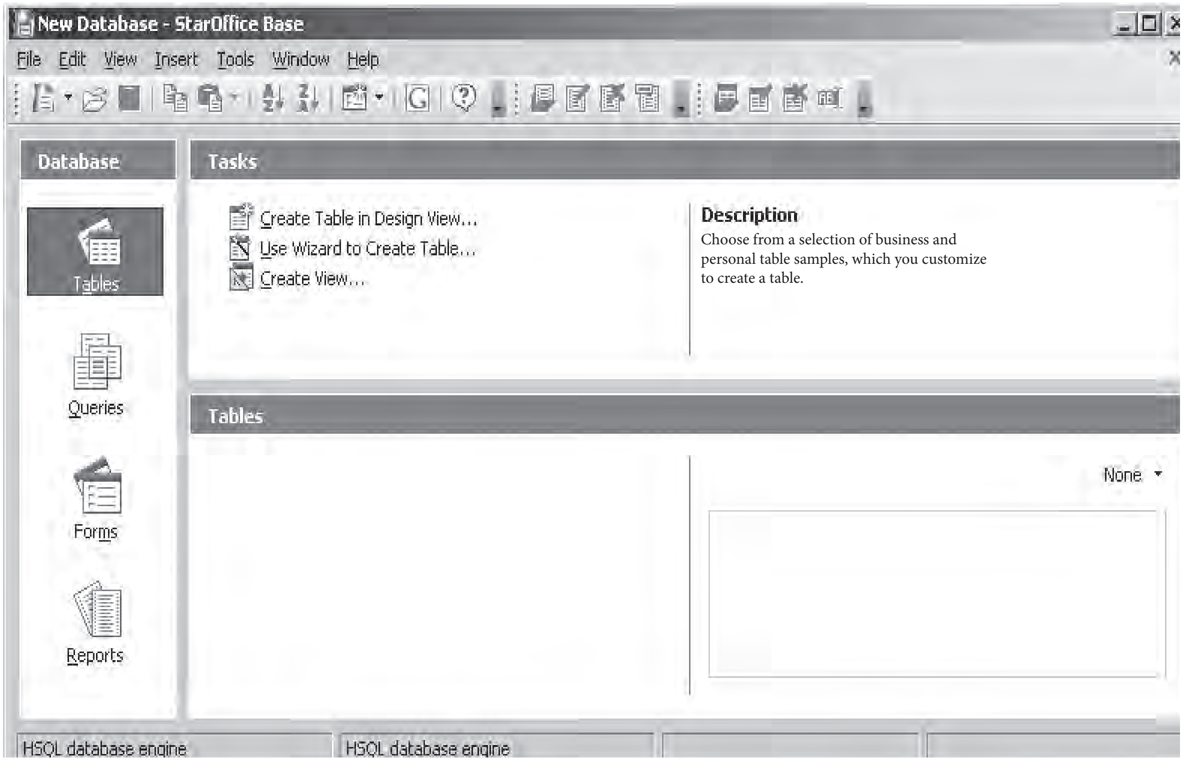
உங்கள் கணிப்பொறியில் **StarOffice** –இன் அமைப்புகள் **(Settings)** வேறுபட்டவையாக இருந்தால், தொடக்கச் சன்னல் திரை படம் 7.1–இல் காட்டியுள்ளதைவிட மாறுபட்டதாக இருக்கும்.

இப்பொழுது தரவுத்தளத்தை எப்படி உருவாக்குவது என்பதைப் பார்ப்போம். **StarOffice Base**–ஐ திறக்க **Start →Programs →StarOffice 8→ StarOffice Base** என்ற பொத்தான்களை கிளிக் செய்க. அல்லது **StarOffice** திரை முன்பே திறந்திருந்தால் **File - New - Database** என்ற பட்டிகளை 'கிளிக்' செய்யவும். உடனே தரவுத்தளத்தின் பண்புகளைக் கொண்ட உரையாடல் பெட்டி **(Database wizard)** படம் 7.2 –இல் காட்டியவாறு திரையில் தோன்றும், இப்பொழுது புது தரவு தளத்தை அமைக்க '**Create a new database**' என்ற இணைப்பைத் தேர்வு செய்யவும். ஏற்கனவே உள்ள கோப்பை திறக்க **'Open an existing database file'** என்ற இணைப்பைத் தேர்வு செய்யவும். பிறகு Finish என்ற பொத்தானைக் கிளிக் செய்யவும்.



படம் 7.2 தரவுத்தள பண்புகளைக் காட்டும் படம்

இப்பொழுது **Save As** உரையாடல் பெட்டி தோன்றி நீங்கள் உருவாக்கிய தரவுத்தளத்தை சேமிக்கத் தூண்டுவதைக் காணலாம். உடனே தரவுத்தளக்கோப்பின் பெயரை கோப்புப் பெயர் உரைப்பெட்டியில் **(File Name text box)** கொடுக்கவும். **Save** பெட்டியை **Click** செய்து தரவுத்தள கோப்பை சேமிக்கச் செய்யவும். பிறகு தரவுத்தளத்தின் பண்புகளைக் கொண்ட சன்னல் திரைப்பகுதிக்கு வந்து **OK** என்ற பட்டியை 'கிளிக்' செய்யவும். உடனே படம் 7.3–இல் காட்டியுள்ள திரை தோன்றுவதைக் காணலாம்.



படம் 7.3 தரவுத்தளத்திற்கு வேண்டிய படிவங்கள் (Forms), வினவல்கள் (Queries), **அறிக்கைகள் (Report), அட்டவணைகள் (Table) இவற்றைத் தேர்வு செய்ய உதவும் திரை.**

சன்னல் திரையின் இடதுபுற சாரளம் **(Pane) Tables, Queries, Forms, Reports** ஆகிய பணிப்பட்டிகளைக் காட்டுவதை நோக்கவும், சாரளத்தின் வலதுப்புற மேற்பகுதியில் பணி விவரங்களும், வலதுப்புறக் கீழ்ப்பகுதியில் மாற்றம் செய்யத் தேர்வு செய்த**Tables, Queries, Forms** அல்லது**Reports** தோன்றும்.

உருவாக்கிய தரவுத்தளத்தில் மேற்கொண்டு பணி செய்ய மேற்கூறிய எந்த பணிக்குறியை வேண்டுமானலும் 'கிளிக்' செய்துத் தொடராலாம்.

செய்து பார்

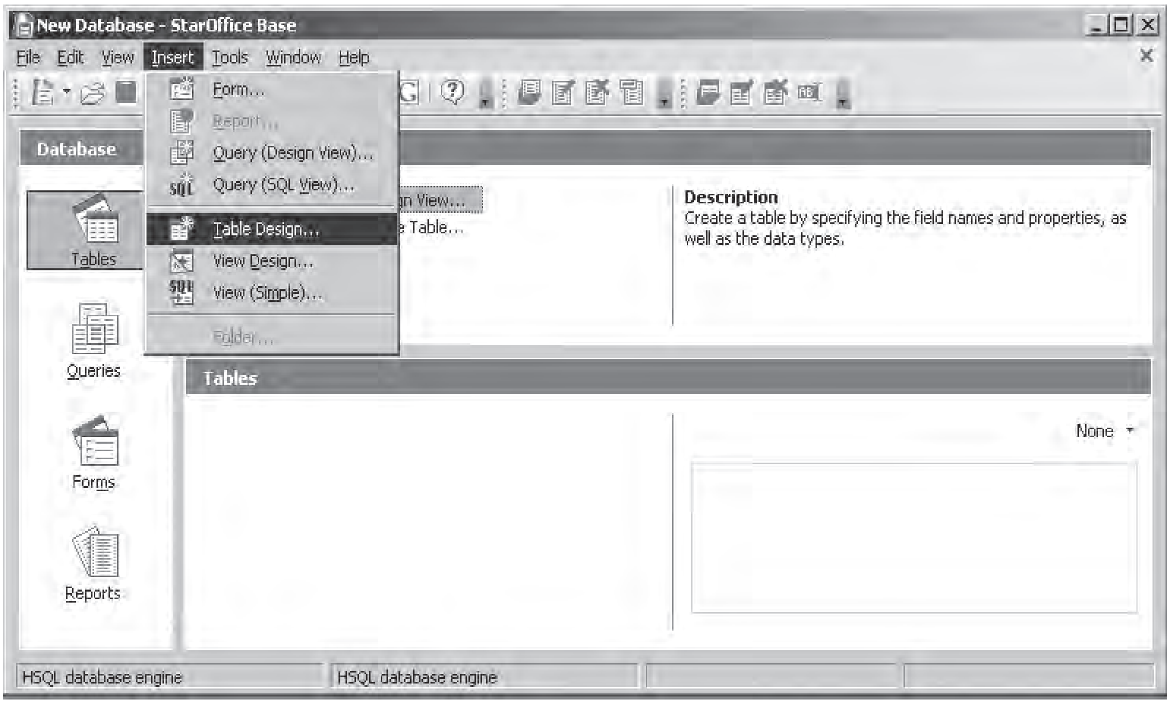
மேற்கூறிய வழிமுறையைப் பயன்படுத்தி 'Student' என்னும் ஒரு தரவுத்தளத்தை உருவாக்கவும்.

7.6.1 உருவாக்கிய தரவுத்தளத்தைக் கையாளல்

முன்பே கூறியது போல, ஒரு தரவுத்தளம் என்பது தொடர்புடைய அட்டவணைகளின் தொகுப்பாகும். தரவுத்தளத்தை உருவாக்கிய பின் அடுத்த பணி, தரவுத்தளத்தில் அட்டவணைகளைஉருவாக்கி, அவற்றில் தரவுகளை உள்ளிடலாகும். இந்த அட்டவணைகளைப் பயன்படுத்தி **Star Base** நம்மை படிவங்கள் **(Forms)** வடிவமைக்கவும், தரவுத்தளத்தை வினாவி விடை காணவும், அறிக்கைகள் உருவாக்கவும் உதவுகின்றது. இவற்றை எப்படிச் செய்வது என்பதை கீழ்வரும் பத்திகளில் படிப்போம்.

7.6.1.1 அட்டவணைகளை உருவாக்கல்

ஒரு புது அட்டணையை உருவாக்க**<Database>** சாளரத்தில் உள்ள **Table** என்ற பட்டியை கிளிக் செய்து, ‘Task’ சாளரப்பகுதியில் உள்ள **‘Create Table in Design view’** என்ற இணைப்பை கிளிக் செய்க. அல்லது **Insert** பட்டியிலிருந்து **‘Table Design’** என்ற பொத்தானைத் தேர்வு செய்யவும். உடனே படம் 7.4 –இல் காட்டியுள்ள திரை தோன்றுவதைக் காணலாம்.



படம் 7.4 புது அட்டவணையை உருவாக்கல்

இப்பொழுது படம் 7.5 –இல் காட்டியுள்ள அட்டவணை வடிவமைக்கும் சன்னல் திரை **(Table Design Window)** தோன்றுவதைப் பார்க்கலாம்.

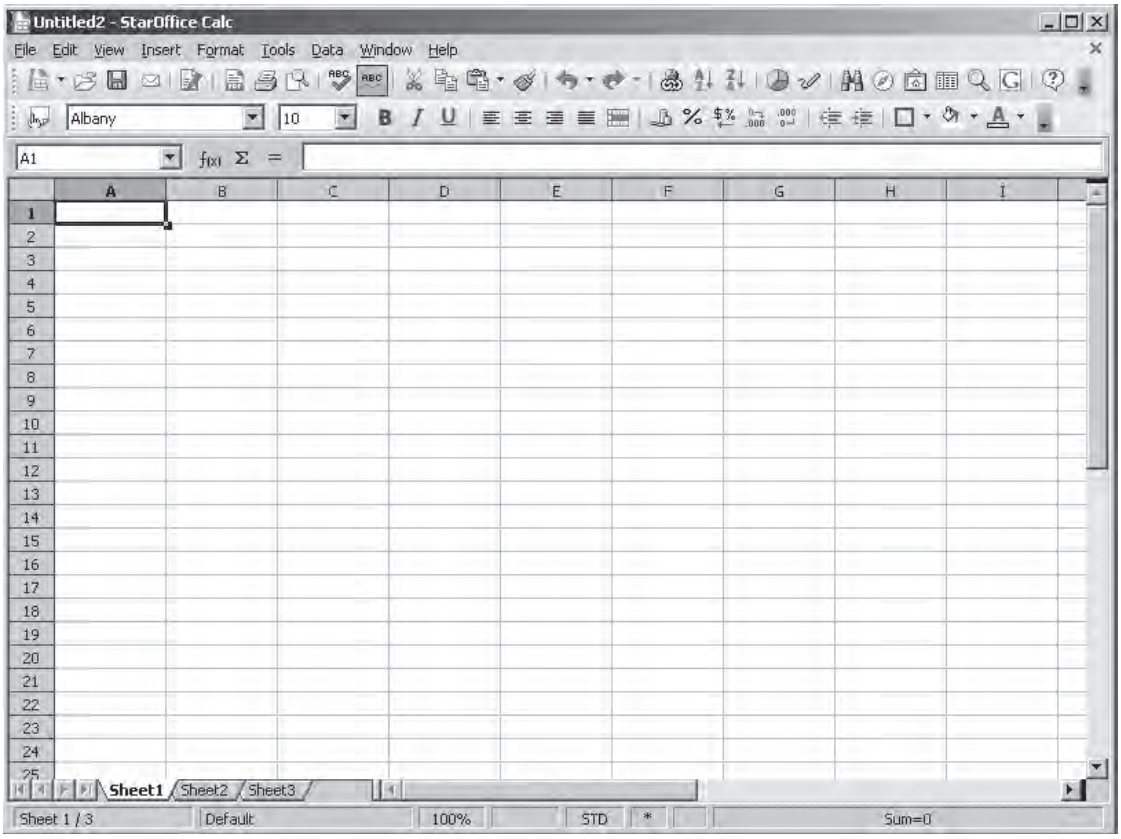
அடுத்து, அட்டவணையின் அமைப்பை **(Structure)** வடிவமைக்க வேண்டும். அட்டவணையை வடிவமைப்பது என்றால், அட்டவணையில் சேர்க்க வேண்டிய வெவ்வேறு புலங்களை **(Fields)** தெரிந்து முடிவு செய்தல், ஒவ்வொரு புலத்திற்கும் பெயரிடல், தரவு வகையை முடிவு செய்தல், போன்ற செயற்பாடுகளைச் செய்வதாகும். ஒவ்வொரு புலத்திலும் கொடுக்க வேண்டிய தரவுகள் பற்றி சிறு குறிப்புகளை வேண்டுமானாலும் கொடுக்கலாம். **Starbase** –இல் **20** வகையான புல வகைகள் **(field types)** உள்ளன. அவையாவன:

1.	உரை (மாறும் உரு)	–	**Text [VARCHAR]**
2.	உரை (மாறும் உரு – தவிர் வகை)	–	**Text [VARCHAR IGNORECASE]**
3.	உரை (மாறா உரு)	–	**Text (fix) [CHAR]**
4.	எண்	–	**Number [NUMERIC]**
5.	தேதி/நேரம்	–	**Date/Time [TIMESTAMP]**
6.	தேதி	–	**Date [DATE]**
7.	நேரம்	–	**Time [TIME]**
8.	ஆம்/ இல்லை	–	**Yes / No [BOOLEAN]**
9.	அறிவிப்பு	–	**Memo [LONGVARCHAR]**
10.	உருவம்	–	**Image [LONGVARBINARY]**
11.	தசமப்புள்ளி எண்கள்	–	**Decimal [DECIMAL]**
12.	இரும எண் புலம் (மாறா)	–	**Binary (fixed) [BINARY]**
13.	இரும எண் புலம்	–	**Binary [VARBINARY]**
14.	முழு எண்	–	**Integer [INTEGER]**
15.	குறு முழு எண்	–	**Tiny Integer [TINYINT]**
16.	சிறு முழு எண்	–	**Small Integer [SMALLINT]**
17.	பெரு முழு எண்	–	**Big Integer [BIGINT]**
18.	மிதவை	–	**Float [FLOAT]**
19.	மெய்மாறி	–	**Real [REAL]**
20.	இரட்டை	–	**Double [DOUBLE]**

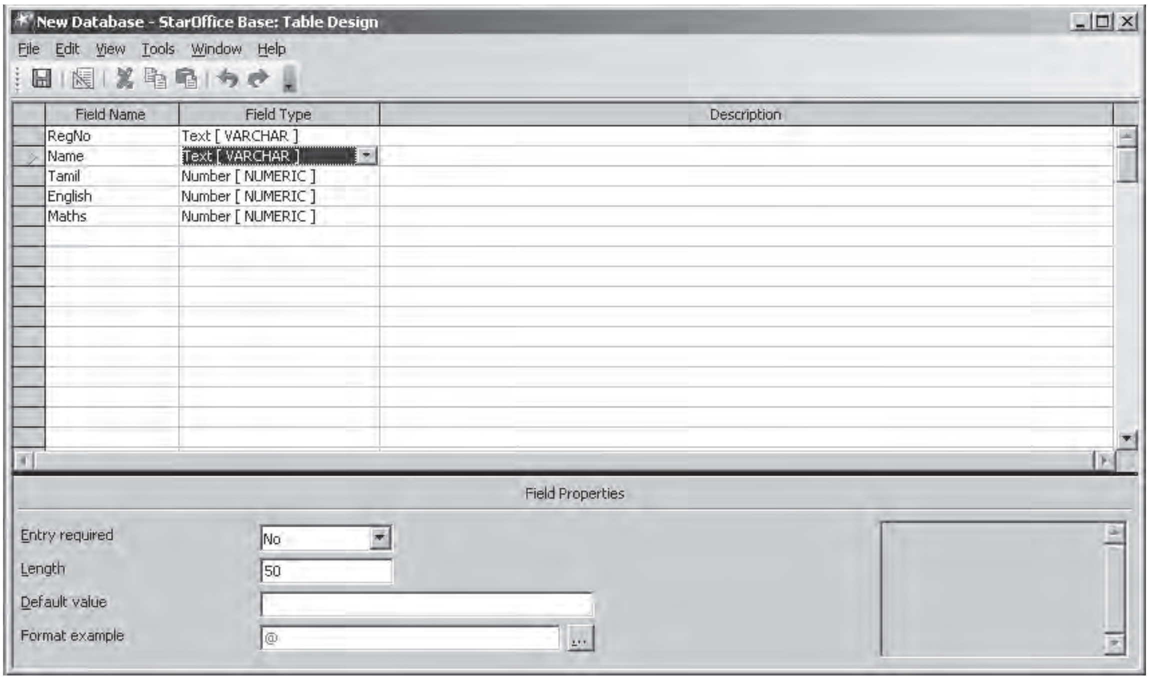
முழு எண் தரவு வகை முழு எண்களை மட்டுமே ஏற்கும். சிறு முழு எண் தரவு வகை சில ஆயிரம் வரை முழு எண்களை ஏற்கும். மிதவை, இரட்டை தரவு வகைகள், மிதவை ஒற்றைத் துல்லியம் **(single precision),** மிதவை இரட்டை துல்லியம் ஆகியவற்றைக் குறிக்கும். ஒரு தசமப்புள்ளிகள் வரை துல்லியமாகக் கொடுக்கவல்லது. மிதவைத் தரவு வகை 14 தசமப் புள்ளிகள் வரை துல்லியமாகக் கொடுக்க வல்லது இரட்டைத் தரவு வகை.

தொடங்கும் பொழுது, சுட்டி புலப்பெயருக்கு உரிய முதல் நெடுவரிசையின் **(First Column)** முதல் புலத்தில் **(First field)** இருக்கும். இப்பொழுது புலத்தின் பெயரை உள்ளிடு. பிறகு தத்தித்திறவு கோலை **(TabKey)** அழுத்தி அடுத்த நெடுவரிசைக்குச் செல்லவும். இந்த நெடுவரிசையில் புலத்தின் வகை **(Field Type)** யைக்கொடுக்க வேண்டும். இதனைச் செய்ய, இங்குள்ள இழுபட்டியல் பெட்டியிலிருந்து **(Drop - down. List box)** வேண்டிய புலவகையை **(field type)** தேர்வு செய்து ‘கிளிக்’ செய். சன்னல் திரையின் கீழ்ப்பகுதியில் புலப்பண்புகளின் சாரளம் **(Field Properties Pane)** உள்ளதைப் பார்க்கலாம். ஒவ்வொரு புலவகைக்கும்

அவற்றின் பண்புகள் காட்டப்பட்டுள்ளதை நோக்கவும், இவற்றைக் கொண்டு புலங்களைப் பயன்பாட்டிற்கு வேண்டியவாறு அமைத்துக் கொள்ளலாம்.



படம் 7.5 அட்டவணையை வடிவமைக்கும் சன்னல்திரை



படம் 7.6 உரைப் புலத்திற்கான புலப்பண்புகள்

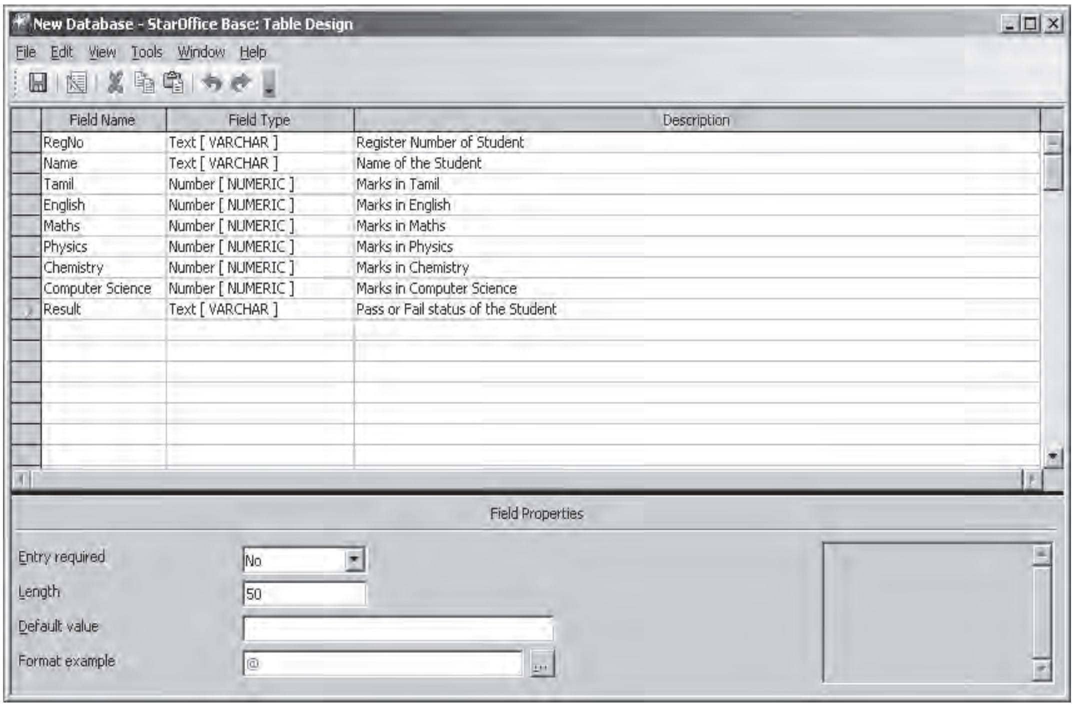
எடுத்துக்காட்டாக தேர்வு செய்த புலத்தின் வகை உரை **(Text)** என்றால், படம் **7.6–**இல் காட்டியுள்ள புலப்பண்புகள் தோன்றும். ஒரு குறிப்பிட்ட புலத்திற்கு தரவுகளை எப்பொழுதுமே பயனார்தான் கொடுக்க வேண்டும் என்று விரும்பினால் இத்திரையில் முதலில் உள்ள **'Entry Required'** என்ற பட்டியைப் பயன்படுத்த வேண்டும். **'Length'** என்ற பட்டி, புலத்தின் அகலத்தை (மிகுதியாக எடுத்துக் கொள்ளக்கூடிய உருக்களின் எண்ணிக்கையை) வரையறுக்கப் பயன்படுகின்றது. **'Default Value'** என்ற பட்டி, ஒரு புலத்திற்கு புலமதிப்புகளை பயனாளர் கொடுக்காத நிலையில் தானே , ஒரு மதிப்பை எடுத்துக்கொள்ளுகின்ற வகையில், தற்கோள் மதிப்பைக் **(Default Value)** கொடுக்கப் பயன்படுகிறது. கடைசியாக உள்ள பட்டி **'Format Example'** பட்டியாகும் இந்தப்பட்டி, எண் வடிவமைப்பையும் **(Number Format)** எண்கள் அல்லது உரையை நெடுவரிசையின் அகலத்திற்குள் இடம், வலம், மையம் என ஒழுங்குபடுத்தும் செயற்பாடுகளையும் செய்ய வேண்டிய அமைப்புகளை **(Settings)** வரையறுக்கப் பயன்படுகிறது.

புல வகை எண்ணாக இருக்கும்போது, புலப்பண்பு **(Field Properties)** பட்டியலில் உள்ள **Length**–ஐ தேர்வு செய்து எண்ணின் நீளத்தை முடிவு செய்து கொள்ளலாம். **'Decimal Places'** –ஐ தேர்வு செய்து எத்தனை தசம எண்கள் தேவையோ அதை முடிவு செய்து கொள்ளலாம். பிற தரவுகளை **Text** தரவு வகை போலவே முடிவு செய்து கொள்ளலாம்.

புலப்பண்புகளை வரையறுத்தபின், தத்தித்திறவுகோலை **(Tab Key)** அழுத்தி அடுத்த நெடுவரிசைக்குச் செல்லவும். இந்த நெடுவரிசையில் புலத்தைப் பற்றிய சில விவரங்களைக் **(Descrptions)** கொடுக்கலாம். இதன் பிறகு தத்திதிறவுகோலை **(Tab Key)** அழுத்தி, இரண்டாவது புலத்திற்குச் செல்லவும். மேலே முதல் புலத்தை வடிவமைக்க விளக்கியதைப் போலவே, மாற்றுமுள்ள எல்லா புலன்களையும் வடிவமைத்து புலப் பண்புகளை வரையறுக்க வேண்டும்.

அடுத்த பணி, முதன்மைத் திறவுகோலை **(Primary Key)** உருவாக்குதலாகும். முதன்மைத் திறவுக்கோல் என்பது, ஒரு அட்டவணையில் உள்ள பதிவுகள் **(Record)** ஒவ்வொன்றையும் அடையாளம் காட்டவல்ல தனித்தன்மை வாய்ந்த ஒருபுலம் அல்லது ஒன்றுக்கு மேல் சேர்ந்த புலங்கள் ஆகும். எடுத்துக்காட்டாக மதிப்பெண் பட்டியலில் ஒவ்வொரு மாணவணைப் பற்றியும் கொடுக்கப்பட்டுள்ள விவரங்களும் ஒரு பதிவு **(Record)** ஆகும். ஒவ்வொரு மாணவனுக்கும் ஒரு பதிவெண் கொடுக்கப்பட்டுள்ளது. ஒரு பதிவெண் ஒரு மாணவனை மட்டுமே குறிக்கும். ஒரு பதிவெண் இரண்டு மாணவர்களைக் குறிக்காது. எடுத்துக்காட்டாக, அட்டவணை **7.1–**இல் பதிவெண் **1001** என்பது அன்பு **A** என்ற ஒரு மாணவரை மட்டுமே குறிக்கிறது. எனவே இந்த எடுத்துக்காட்டில் பதிவெண் என்ற புலத்தை ஒரு முதன்மைத் திறவு கோலாக **(Primary Key)** வரையறுக்கலாம். இரண்டிற்கும் மேற்பட்ட அட்டவணைகள் கொண்ட தரவுத்தளங்களில், பல அட்டவணைகளுக்கும் பொதுவாக புலங்கள் இருக்கும்போது, முதன்மைத் திறவுகோல் **(Primary Key)** மிக முக்கியமான இடத்தைப் பெறுகின்றது. தரவுத்தளத்தில் உள்ள ஒவ்வொரு அட்டவணைக்கும் ஒரு முதன்மைத் திறவுகோல் தேவை. இந்த எடுத்துக்காட்டில் மாணவர் மதிப்பெண் அட்டவணைக்கு 'பதிவெண்' என்ற புலத்தை முதன்மைத்திறவு கோலாக வரையறுக்க, பதிவெண் **(Reg No)** என்ற புலத்திற்கு இடப்புறத்தில் உள்ள சிறுமுக்கோணக்குறி மீது சுட்டியை வைத்து சுட்டெலியின் வலப்புற பொத்தானை அழுத்தவும். ஒரு குறுபட்டியல் **(Short Menu)** தோன்றும். அதில் **Primary Key** என்ற பட்டியை 'கிளிக்' செய்யவும். இப்பொழுது **Reg No** என்ற புலம் முதன்னைத் திறவுகோல் என்பதை அடையாளம் காட்ட ஒரு சிறு மஞ்சள் நிற பொத்தான் **Reg No** என்ற புலத்திற்கு இடப்புறத்தில் தோன்றுவதைப் பார்க்கலாம்.

இப்படி உருவாக்கிய அட்டவணையின் அமைப்பை ஏதேனும் ஒரு பெயரில் சேமித்துத் தேக்கி வைக்க வேண்டும். இதனை விசைப் பலகையில் உள்ள **Ctrl + S** பொத்தான்களை அழுத்தியோ அல்லது **File** பட்டியில் உள்ள **Save** பட்டியை ‘கிளிக்’ செய்தோ, செயற்படுத்தலாம். படம் 7.7 –இல் மதிப்பெண் பட்டியலுக்கு உருவாக்கிய அட்டவணைக் காட்டப்பட்டுள்ளது. இந்த அட்டவணையை **Mark List** என்ற பெயரில் சேமிக்கவும்.



படம் 7.7 Mark List அட்டவணை வடிவமைப்பு

செய்து பார்

தேதி, பணக்குறியீடு போன்ற தரவு வகைகளை உரை **(Text)**, எண் **(Number)** தரவு வகைகளுடன் பயன்படுத்திப் பழக, கீழ்க்கண்ட பணியாளர் அட்டவணையை வடிவமைக்கவும்

Field Name	Type	Description
EmpNo	Number	Employee Number
StaffName	Text	Name
Qualification	Text	Qualification
Designation	Text	Designation
Department	Text	Department
D-of-Join	Date	Data of joining
MonSal	Currency	Monthly Salary

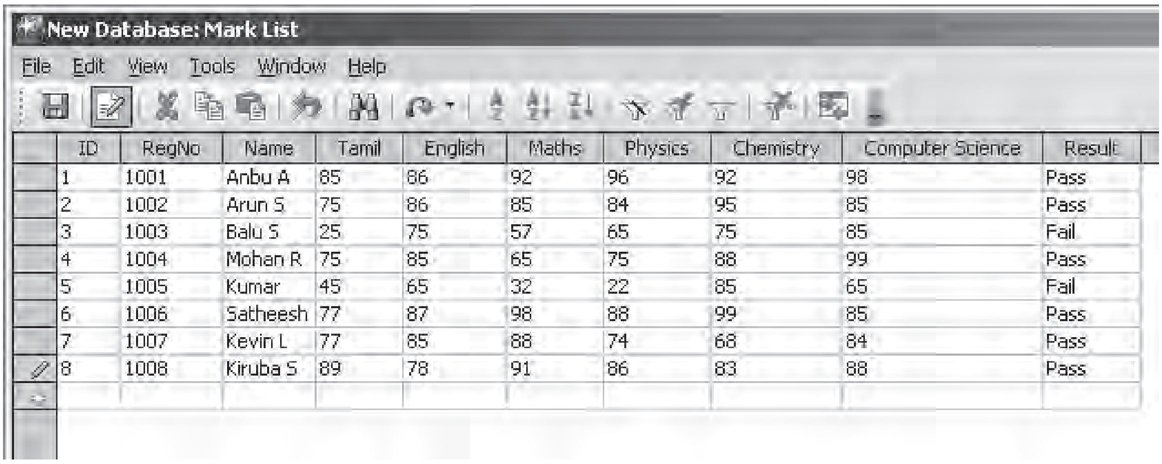
7.6.1.2 தரவுகளை உள்ளிடல் (Entering the Data)

முன்பே வடிவமைத்து **Mark List** என்ற பெயரில் சேமித்து வைக்கப்பட்டுள்ள அட்டவணையைத் திறக்க, **Explorer** சன்னல் திரையில் **Table** என்ற பட்டியின் இடப்புறம் உள்ள **+** என்ற குறியை கிளிக் செய்து, பிறகு **Mark List** என்ற பட்டியை ‘கிளிக்’ செய்யவும். இந்த அட்டவணையை

(Database) சன்னல் திரையில் உள்ள **Mark List** என்ற பட்டியை இருமுறை கிளிக் செய்தும் திறக்கலாம். அட்டவணையைத் திறந்த பின், தரவுகளைக் கொடுக்கலாம்.

சுட்டி, முதல் பதிவின் முதல் புலத்தில் இருப்பதைப் பார்க்கவும். இப்பொழுது, தத்தித்திறவுகோலைக் **(Tab Keys)** கொண்டு ஒவ்வொரு புலமாகச் சென்று தரவுகளைக் கொடுக்கவும். தரவுகளைக் கொடுக்கும் போது, ஒவ்வொரு புலத்திலும் அந்த புலத்திற்கு ஏற்கனவே வரையறுத்த அதே தரவு வகைத் தரவுகளைத்தான் கொடுக்கின்றோமா என்று எச்சரிக்கையுடன் கவனித்துத் தரவுகளைக் கொடுக்க வேண்டும். தவறான தரவு வகைத் தரவுகளைக் கொடுத்தால், உடனே உரையாடல் பெட்டி தோன்றி, கொடுத்துள்ள தரவு வகை சரியில்லை என்று கூறும்.

MarkList அட்டவணை, சில மாதிரித் தரவுகளுடன், படம் 7.8–இல் காட்டப்பட்டுள்ளது.



ID	RegNo	Name	Tamil	English	Maths	Physics	Chemistry	Computer Science	Result
1	1001	Anbu A	85	86	92	96	92	98	Pass
2	1002	Arun S	75	86	85	84	95	85	Pass
3	1003	Balu S	25	75	57	65	75	85	Fail
4	1004	Mohan R	75	85	65	75	88	99	Pass
5	1005	Kumar	45	65	32	22	85	65	Fail
6	1006	Satheesh	77	87	98	88	99	85	Pass
7	1007	Kevin L	77	85	88	74	68	84	Pass
8	1008	Kiruba S	89	78	91	86	83	88	Pass

படம் 7.8 Mark List அட்டவணை

7.6.1.3 தரவுகளைப் பதிப்பாய்தல் (Editing the Data)

பதிப்பாய்வு **(Editing)** என்ற செயலாக்கம், ஒரு பதிவின் புலத்தில் உள்ள தரவுகளை மாற்றுதல், ஒரு பதிவை நீக்குதல், ஒரு பதிவை சேர்த்தல் போன்ற செயற்பாடுகளைக் கொண்டுள்ளது. இத்துடன் **Star calc** எழுத்துக்களின் வகை, அளவு **(Font type and Size)** போன்றவற்றை மாற்றவும் வரிசையின் உயரத்தை மாற்றி அமைக்கவும் வழி செய்கின்றது.

அட்டவணையின் தரவுகளைப் பதிப்பாய்வு செய்தல் மிகவும் எளிது. எந்த புல மதிப்பை மாற்ற வேண்டுமோ அந்த புலத்தின்மேல் சுட்டியை வைத்து ‘கிளிக்’ செய்துவிட்டு புது மதிப்பை (தரவைக்) கொடுக்க வேண்டும். எடுத்துக்காட்டாக, மோகன் **R** என்ற மாணவரின் ஆங்கில பாட மதிப்பெண்ணை 85 –லிருந்து 75 ஆக மாற்ற சுட்டியை **English** என்ற புலத்தில் நான்காவது பதிவில் வைத்து ‘கிளிக்’ செய்துவிட்டு புதிய, எண்ணாகிய 75 –ஐ உள்ளிட வேண்டும்.

பதிவை **(Record)** பதிப்பாய்வு செய்ய, முதலில் பதிவைத் தேர்வு செய்ய வேண்டும். பதிவுச் சுட்டியை **(Record Pointer),** குறிப்பிட்ட பதிவு மேல் வைத்து ‘கிளிக்’ செய்தால் பதிவு, தேர்வு செய்யப்படும். பதிவுச்சுட்டி **(Record Pointer)** என்பது அட்டவணையின் இடப்பக்கக் கடைசி நேர்வரிசையில் **(Cloumn)** உள்ள ஒரு சிறு முக்கோணமாகும். நீங்கள் எந்த பதிவை பதிப்பாய்வு செய்ய வேண்டுமோ அந்தப் பதிவின் பதிவுச்சுட்டியை ‘கிளிக்’ செய்யவும். இப்பொழுது ஒரு குறு பட்டியல் **(Short - Cut Menu)** படம் 7.9– இல் காட்டப்பட்டுள்ளதைப் போல தோன்றும்.

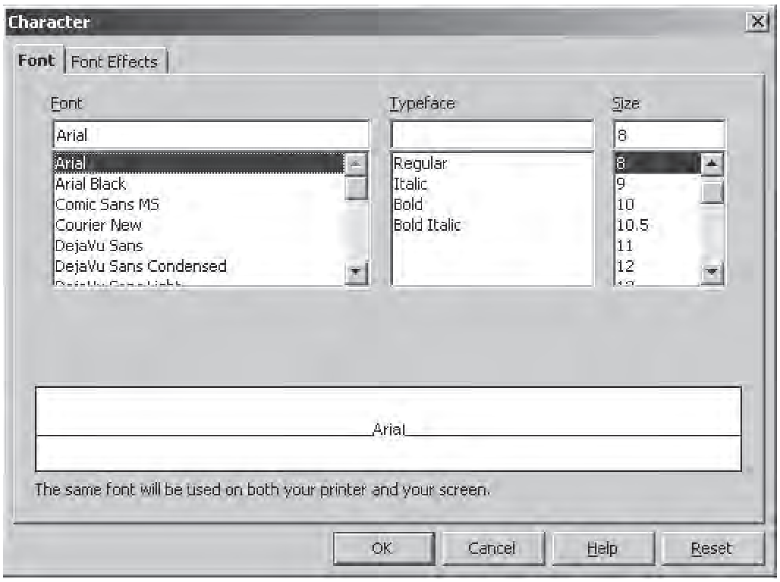
படம் 7.9 பதிப்பாய்வின் குறு பட்டியல் (கிளைப்பட்டியல்)

இந்த பட்டியலைப் பயன்படுத்தி ஒரு பதிவை நீக்கவோ, பதிவின் உயரத்தை மாற்றவோ அல்லது அட்டவணையை மாற்றி வடிவமைக்கவோச் செய்யலாம்.

Delete Rows என்ற பட்டியை தேர்வு செய்து 'கிளிக்' செய்தால் தேர்வு செய்யப்பட்ட பதிவு நீக்கப்படும். ஒரு பதிவை **(Record)** நீக்க வேண்டுமெனில், அந்தப் பதிவைத் தேர்வு செய்து **'Delete Rows'** பட்டியை 'கிளிக்' செய்யவும். உடனே உரையாடல் பெட்டி ஒன்று தோன்றும். அதில் உள்ள **'Yes'** பொத்தானை கிளிக் செய்தால் பதிவு நீக்கப்படும். பதிவின் உயத்தை மாற்ற **Row Height** என்ற பட்டியைத்தேர்வு செய்யவும். படம் 7.10–இல் காட்டியுள்ளவாறு ஒரு உரையாடல் பெட்டி **(Dialog Box)** தோன்றும். இதில் தேவையான உயரத்தைக் கொடுத்து **OK** என்ற பட்டியை 'கிளிக்' செய்யவும். மாற்றம் நிகழ்வதைப் பார்க்கலாம்.

படம் 7.10 உயரத்தை நிறுவ உதவும் உரையாடல் பெட்டி

அட்டவணையை வடிவமைக்க விரும்பினால், **Table Format** பட்டியை 'கிளிக்' செய்யவும். படம் 7.11 –இல் காட்டியுள்ள உரையாடல் பெட்டி தோன்றும் இதனைப் பயன்படுத்தி எழுத்தின் வகை,



படம் 7.11 அட்டவணை வடிவமைக்கும் சன்னல் திரை

அளவு, நிறம் போன்றவற்றை மாற்றி அமைக்கலாம். ஒரு பதிவைச் சேர்க்க சுட்டியைக் கடைசி வரிசையில் வைத்துக் ‘கிளிக்’ செய்து புது பதிவை உள்ளிடவும் எடுத்துக்காட்டாக, கீழ்க்கண்ட இரு பதிவுகளை **Mark list** அட்டவணையில் சேர்க்கவும்

அ) **Tamilselvan S**

மதிப்பெண்கள்

Tamil: 55,English:68

Maths :95, Physics :75

Chemistry:69, Comp, Sc : 98

Result : Pass

ஆ) **Prem**

மேற்கண்ட பாடங்களில் எடுத்த மதிப்பெண்கள் முறையே

55, 67, 67, 78, 56, 45.

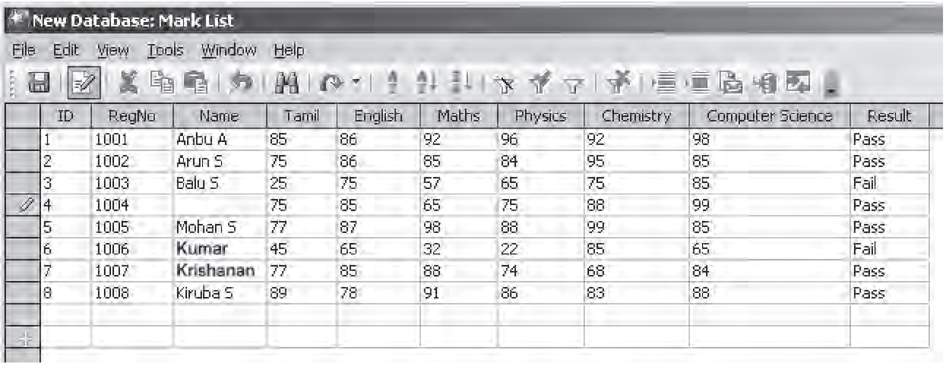
Result : Pass

இதனைச் சேர்க்க, சுட்டியை அட்டவணையின் கடைசி வரிசையில் வைத்து ‘கிளிக்’ செய்து தரவுகளைக் கொடுக்கவும். தரவுகளைக் கொடுக்கும் போது புதுவரிசை தானே தோன்றுவதைப் பார்க்கலாம்.(படம் 7.12)

திட்டமிட்ட மாற்றங்களையெல்லாம் செய்தபின் அட்டவணையை **Ctrl+S** என்ற பொத்தாண்களை அழுத்தி, அட்டவணையை சேமிக்கவும்.

7.6.1.4 அட்டவணை வடிவமைப்பை மாற்றலும் காணலும்

ஏற்கனவே வடிவமைக்கப்பட்டு சேமித்து வைக்கப்பட்டுள்ள அட்டவணையை திரையில் பார்க்கவும், மாற்றி அமைக்கவும் தரவுத்தள சாளரத்தில் **(Database Pane)** உள்ள **Tables** என்ற பட்டியை ‘கிளிக்’ செய்யவும். மாற்றியமைக்க வேண்டிய அட்டவணையைத் தேர்வு செய்து சுட்டெலியின் வலப்புற பொத்தானை அழுத்தவும். ஒரு கிளைப்பட்டியல் தோன்றும். அதில் **Edit** என்ற பட்டியைத் தேர்வு செய்யவும். ஏற்கனவே இருக்கின்ற அட்டவணையின் ஒரு சன்னல் திரையில் தோன்றும். இங்கே, புலப்பெயர், புலவகை, புலப்பண்புகள் இவற்றை மாற்றி நிறுவுதல், புதுபுலத்தைச் சேர்த்தல், உள்ள புலத்தை நீக்குதல் போன்ற மாற்றங்களைச் செய்யலாம். மாற்றங்களைச் செய்தபின் மாற்றியமைக்கப்பட்ட அட்டவணையை சேமித்துக் கொள்ளலாம்.



ID	RegNo	Name	Tamil	English	Maths	Physics	Chemistry	Computer Science	Result
1	1001	Anbu A	85	86	92	96	92	98	Pass
2	1002	Arun S	75	86	85	84	95	85	Pass
3	1003	Balu S	25	75	57	65	75	85	Fail
4	1004		75	85	65	75	88	99	Pass
5	1005	Mohan S	77	87	98	88	99	85	Pass
6	1006	Kumar	45	65	32	22	85	65	Fail
7	1007	Krishanan	77	85	88	74	68	84	Pass
8	1008	Kiruba S	89	78	91	86	83	88	Pass

படம் 7.12 பதிப்பாய்வு செய்யப்பட்ட மதிப்பெண் அட்டவணை

செய்து பார்

Mark list அட்டவணையில் கீழ்க்கண்ட மாற்றங்களைச் செய்.

1. **Tamil, English** என்ற புலங்களின் புலவகையை உரைப்புலம் **(Text field)** என்று மாற்றி மதிப்பெண்களை கீழ்க்கண்ட எழுத்துவகைத் தரநிலைகளாக **(Grades)** மாற்றி அமைக்க.

மதிப்பெண் வீச்சு	தரநிலை **(Grade)**
85 -100	H
75 - 85	A+
65 - 75	A
55 - 65	B+
50 - 55	B
45 - 50	C+
40 - 45	C
< 40	F

2. அட்டவணையில் **'Comments'** என்ற புதுப்புலத்தைச் சேர்த்து, ஒவ்வொரு மாணவர் பதிவிலும் கீழ்க்கண்டவாறு சான்று தருக.

H & A+	**Outstanding**
A & B+	**Excellent**
B & C+	**Good**
C	**Fair**
F	**Poor**

7.6.1.5 பதிவுகளை வரிசைப்படுத்துதல் (Sorting the Records)

அட்டவணையில் கொடுக்கப்பட்டுள்ள பதிவுகளை திரும்பவும் ஒழுங்குபடுத்தி அமைக்க **StarBase** வழி செய்கின்றது. ஏதேனும் ஒரு புலத்தில் உள்ள தரவுகளை ஏறு வரிசையிலோ, இறங்கு வரிசையிலோ வரிசைப்படுத்தி அதற்கேற்ப பதிவுகளை மாற்றி அமைக்க முடியும். இதனைச் செயல்படுத்த கீழ்க்கண்டவாறு செய்க:

1. வரிசைப்படுத்த விரும்பும் அட்டவணையை திற.

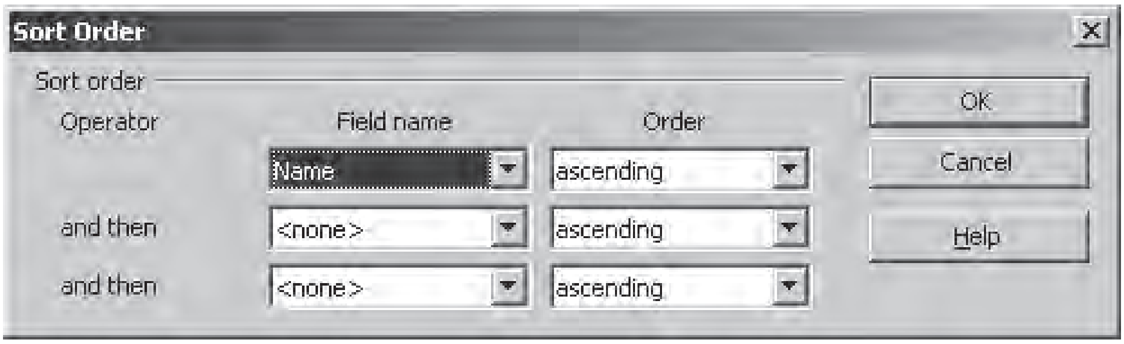
2. வரிசைப்படுத்த விரும்பும் புலத்தைத் தேர்வு செய் (அட்டவணையின் உச்சியில் உள்ள புலப்பெயரை கிளிக் செய்.) அந்த நெடுவரிசை முழுவதும் மிகைப்படுத்தப்பட்டு மிளிர்வதைக் கவனி, பிறகு **Sort Ascending** என்ற பணிக்குறியை 'கிளிக்' செய். இப்பொழுது அட்டவணையின் பதிவுகள், குறிப்பிட்ட புலத்தின் ஏறு வரிசைக்கேற்ப ஒழுங்கு படுத்தப்பட்டு இருப்பதைப் பார்க்கலாம்.

3. இறங்கு வரிசையில் ஒழுங்கு படுத்த புலத்தைத் தேர்வு செய்து **Sort Descending** என்ற பணிக்குறியை 'கிளிக்' செய்க.

- **பற்புல வரிசையாக்கம் (Multiple Sorting)**

பற்புல வரிசையாக்கம் என்றால் ஒரே நேரத்தில் ஒரு அட்டவணையின், ஒன்றிற்கும் மேற்பட்ட புலத்தரவுகளை வரிசைப்படுத்தி, பதிவுகளை ஒழுங்குபடுத்தும் செயலாக்கமாகும். எடுத்துக்காட்டாக, **Mark list** அட்டவணையை (படம் 7.13) எடுத்துக் கொள்வோம். இப்பொழுது பெயர் புலத்திலுள்ள பெயர்களை அகர வரிசைப்படுத்தி அமைக்க விரும்புவதாகக் கொள்வோம். இதனைச் செயல்படுத்த கீழ்க்கண்டவாறு செய்க:

1. **Sort** பணிக்குறியை 'கிளிக்' செய் படம் 7.13 உள்ளதைப் போன்ற உரையாடல் பெட்டி தோன்றும்.



படம் 7.13 **Sort Order Window**

2. **Field Name** இழு பட்டியல் பெட்டியிலிருந்து, **(Drop - Down List Box)** வரிசைப்படுத்த விரும்பும் புலங்களைத் தேர்வு செய்து குறிப்பிடு. தேர்வு செய்த புலங்கள் ஒவ்வொன்றும் எந்த முறையில் வரிசைப்படுத்த வேண்டும் என்பதை **Order** இழுபட்டியலில் **(Drop - Down List Box)** குறிப்பிடு. இதனைச் செய்தபின் **OK** பட்டியை 'கிளிக்' செய். பதிவுகள் விரும்பியவாறு வரிசைப்படுத்தப்பட்டுத் திரையில் தோன்றுவதைப் பார்க்கலாம்.

பதிவுகளை பழையபடி பார்க்க, விரும்பினால் **Remove Filter / Sort** பணிக்குறியை 'கிளிக்' செய்.

செய்து பார் பயிற்சி

Mark list அட்டவணையை, பெயர்களின் அகர வரிசையில் ஏறு வரிசைப்பட்டியும், பதிவெண்களை கீழ்வரிசைப்படியும் ஒழுங்குபடுத்திக் காட்டுக.

7.6.2 தரவுத்தளத்தை வினவுதல் (Querying a Data Base)

ஒவ்வொரு தரவுத்தள மேலாண்மை அமைப்பும் **(DBMS)** கணிப்பொறியின் நிரல் மொழியைப் **(Programming Language)** போன்ற ஒரு மொழியை ஆதரிக்கின்றது. **DBMS**க்குத் துணை நிற்கும் இந்த மொழியை **Structured Query Language (SQL)** "வினவல் அமைப்பு மொழி" என்று அழைக்கின்றனர். கணிப்பொறியின் நிரல் மொழிகளை விட மிக எளிய முறையில் தரவுத் தளங்களுடன் தொடர்புகொண்டு செயல்பட இந்த மொழி பயன்படுகின்றது. வினவல்

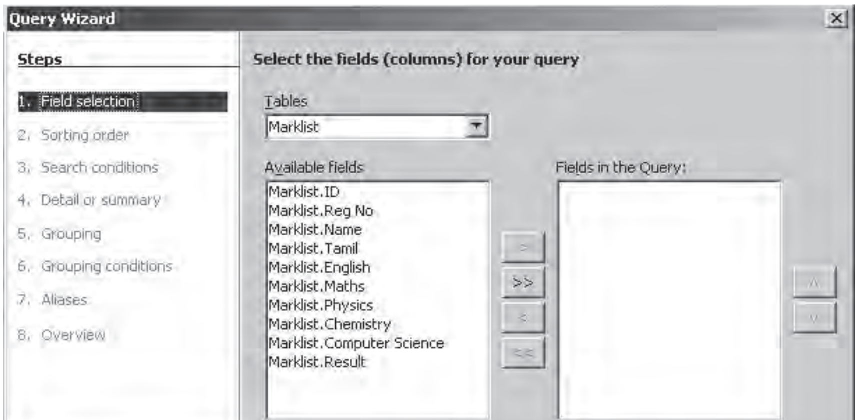
மொழியைப் **(Query Language)** பயன்படுத்தி கீழ்க்கண்ட செயற்பாடுகளைச் செய்ய முடியும்.

- தரவுத்தளங்களில் பதிவுகளைத் தேடிக் கண்டுபிடித்தல்
- பதிவுகளை நிகழ்நிலைப்படுத்த **(Update)** வேண்டி, அட்டவணைக்கு இடையே உறவுகள் **(Relationships)**அல்லது இணைப்புகளை **(Link)** ஏற்படுத்துதல்
- பதிவுகளின் ஒரு துணைத் தொகுதியைப் **(Subset)** பட்டியலிடல்
- கணக்கீடுகளைச் செய்தல்
- வேண்டாத பதிவுகளை நீக்குதல்
- பிற தரவுத்தள மேலாண் செயற்பாடுகளைச் செய்தல்

வினவல்கள் **(Queries)** என்பன. ஒரு அட்டவணைத் தரவுகளின் சிறப்பு நோக்குகளாகும் **(Special Views)**. வரிசையாக்கம் **(Sort)** போல் வினவல் **(Query)** செயற்பாடு, மூல அட்டவணையை மாற்றுவதில்லை. வினவலின் விளைவுகள் எப்பொழுதும் தனியே வேறு நினைவாகப் பகுதியில் தேக்கி வைக்கப்படுகின்றன. எனவே ஒரு முறை செயற்படுத்திய வினவல்களின் விளைவுகளை, எப்பொழுது வேண்டுமானாலும் பார்த்துக் கொள்ளலாம்.

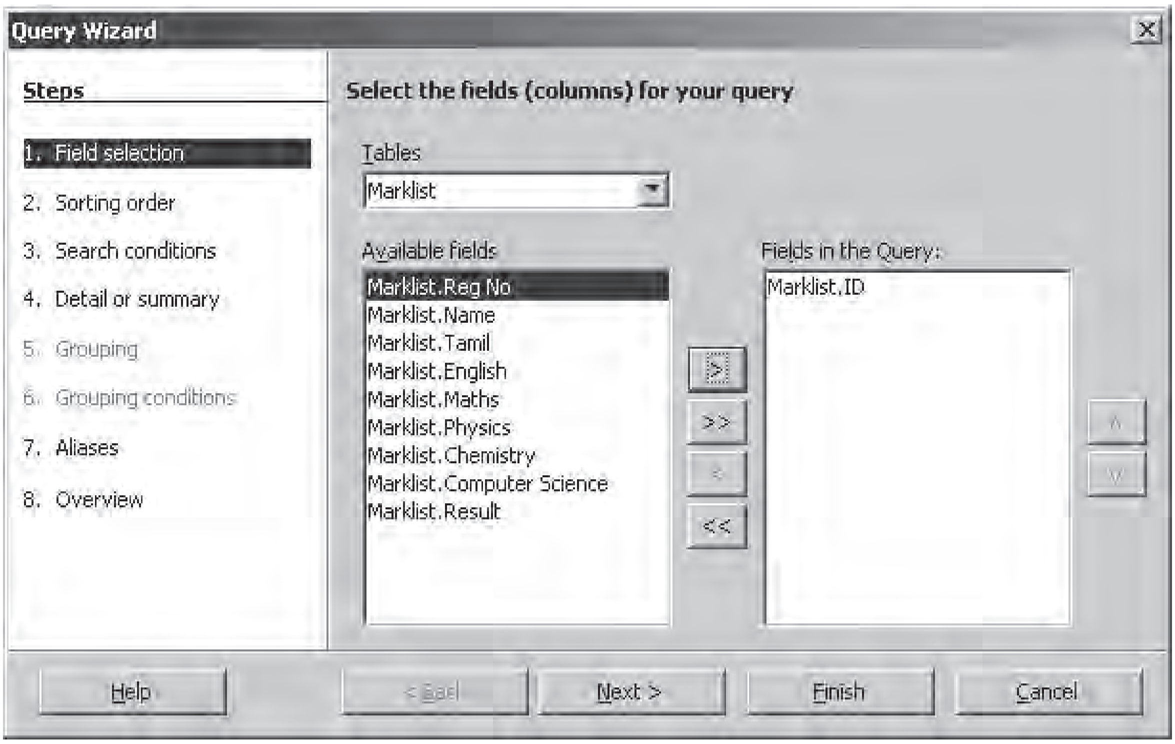
Star Base–இல் ஒரு வினவலை **(Query)** உருவாக்க, தரவுத்தள சாளரத்தில் உள்ள **Queries** என்ற பட்டி மேல் 'கிளிக்' செய்யவும், படம் 7.14–இல் உள்ளதைப் போல **'Autopilot'**–இன் வினவல் சன்னல் திரை தோன்றும்.

Task சாளரத்தில் **'Create Query in Design View'** , **'Use Wizard to Cerate Query'** மற்றும் **'Create Query in SQL View'** என்ற இணைப்புகள் தோன்றும். **'Use Wizard to Cerate Query'** இணைப்பில் 'கிளிக்' செய்தால் படம் 7.14 கொடுக்கப்பட்டுள்ள திரை தோன்றும்.



படம் 7.14 Query Wizard வினவல் சன்னல் திரை

இந்தத் திரையின் **Tables** சுழல் பெட்டியில் ஏற்கனவே உருவாக்கப்பட்டுள்ள அட்டவணைகளின் பட்டியல் கொடுக்கப்பட்டிருக்கும். இந்தப் பட்டியலிலிருந்து, எந்த அட்டவணைப் பற்றி வினவல்கள் எழுப்ப விரும்புகின்றீர்களோ, அதனைத் தேர்வு செய்யவும். பிறகு சன்னல் திரையில் தேர்வு செய்யப்பட்ட அட்டவணையில் உள்ள புலங்களின் பட்டியல் **'Available Fields'** உரைப் பகுதியில் தானே தோன்றும். (படம் 7.15)



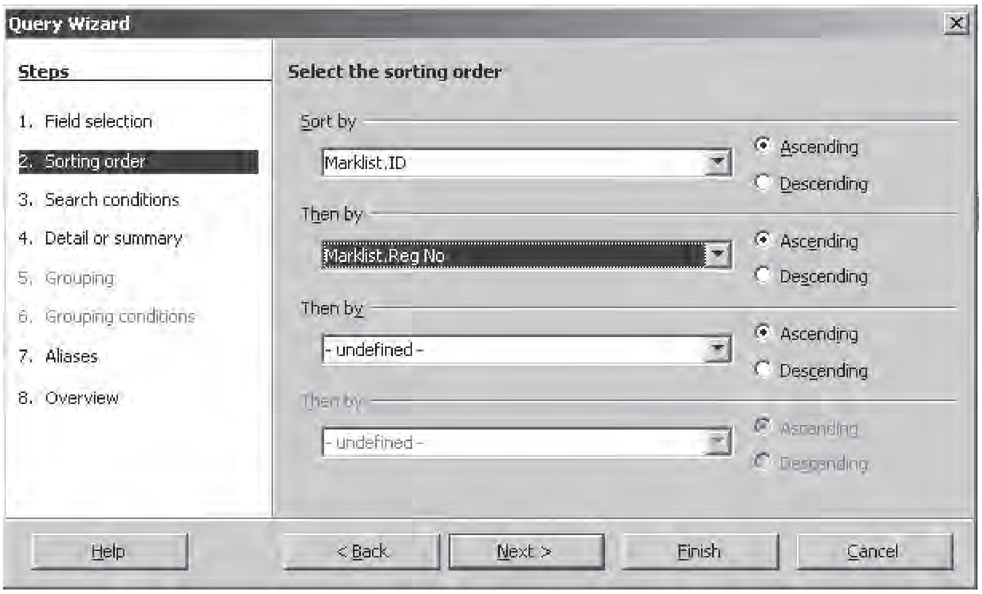
படம் 7.15 புலத்தேர்வு சன்னல் திரை

வினவலில் சேர்க்க வேண்டிய புலங்களைத் தேர்வு செய்ய,**'Available Fields'** உரைப் பகுதியில் உள்ள புலத்தை 'கிளிக்' செய்து **>** பொத்தானை 'கிளிக்' செய்யவும். எல்லா புலங்களையும் ஒரே நேரத்தில் தேர்வு செய்ய **>>** பொத்தானைக் 'கிளிக்' செய்க. பிறகு **Next** என்ற பொத்தானை 'கிளிக்' செய். படம் 7.16– இல் உள்ள திரை தோன்றும்.

வரிசைப்படுத்த விரும்பும் நான்கு புலங்களைத் தேர்வு செய்து செயலாற்ற இந்த சன்னல் திரை பயன்படுகிறது. (படம் 7.16)

வலப்புற உள்ள பொத்தான் மூலம் ஏறுவரிசை அல்லது கீழ்வரிசையைத் தேர்வு செய்து கொள்ளலாம். வரிசைப் படுத்தும் முறையைத் தேர்வு செய்ய வேண்டும் என்ற கட்டாயம் இல்லை. **Next** பொத்தானை நேரடியாக 'கிளிக்' செய்து கொள்ளலாம். அவ்வாறு செய்தால் ஏறு வரிசையைத் தானே தேர்வு செய்து கொள்ளும். இப்பொழுது **Next** பொத்தானை 'கிளிக்' செய்யுங்கள். கீழ்க்காணும் திரை தோன்றும். இந்தத் திரை புலங்களைத் தேர்வு செய்யவும் வினா விவரங்களைக் கொடுக்கவும் பயன்படுகிறது. எடுத்துக்காட்டாக, தேறிய மாணவர்களின் பெயர்களைப் பட்டியலிட விரும்புவதாகக் கொள்வோம். இதனைச் செயல்படுத்த கீழ்க்கண்டவாறு செய்க.

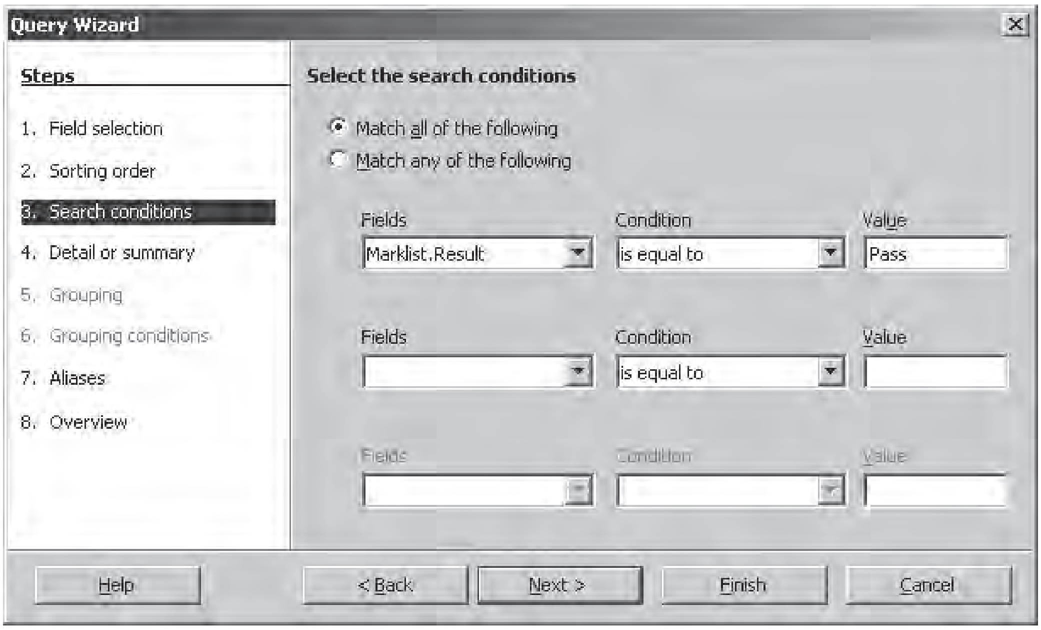
- Result புலத்தை **Field Name** பெட்டியிலிருந்து தேர்வு செய்க.
- பிறகு **Condition –** பெட்டியிலிருந்து **'is equal to'** என்ற பொத்தானைத் தேர்வு செய்க.
- அதன் பின் **Value** பெட்டியில் **Pass** என்று கொடு
- **'Finish'** பொத்தானை 'கிளிக்' செய். இப்பொழுது **Result** புலத்தில் **'Pass'** என்றுள்ள அனைத்து மாணவர்களின் பதிவுகளும் திரையில் தோன்றும்.



படம் 7.16 **Sort Order** சன்னல் திரை

இது ஒரு எளிய வினவலாகும். **Condition** பெட்டியில் உள்ள **<>, <, <=, >=....** போன்ற உறவு நிலை வினைக்குறி **(Relational Operators)** களை பயன்படுத்தி வினவல்களை **(Queries)** உருவாக்கலாம்.

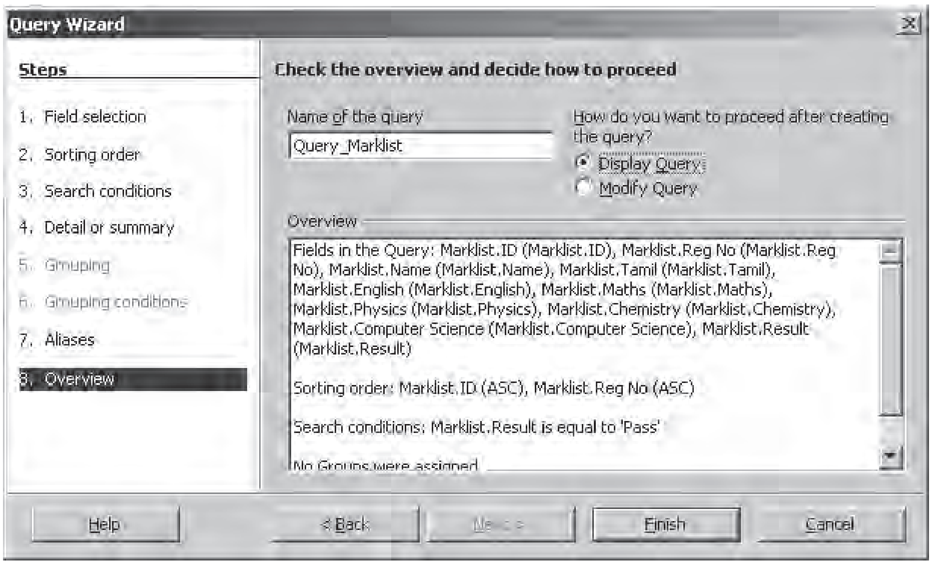
ஏதேனும் ஒரு தேடுநிலையை நிறைவு செய்யும் பதிவுகளை திரையிலிட வேண்டுமெனில், **'Match any of the following'** அல்லது **'Match all of the following'** என்ற இணைப்பைத் தேர்வு செய்யவும்.



படம் 7.17 Search Order சன்னல் திரை

Next பொத்தானை 'கிளிக்' செய்யவும். **Query Wizard - Detail or Summary** தோன்றும். **(figure 7.17)** இது வினாவின் அனைத்து பதிவுகளையும் திரையிலிட வேண்டும் அல்லது விடையின் தொகு கோவை **(agreegate functions)** களை மட்டும் திரையிலிட வேண்டுமா

என்பதை வரையறுக்கும். இந்தத் திரை வினாவில் எண் புலங்கள் இருந்தால் மட்டும் தொகு கோவைகளைத் திரையிட வழி செய்யவும். பிறகு **Next** பொத்தானை 'கிளிக்' செய்யவும். **Query Wizard** இன் **Aliases** திரை தோன்றும்.



படம் 7.18 Create வினவல் சன்னல் திரை

இந்த சன்னல் திரை, வினவலுக்கு ஒரு பெயர் கொடுக்க வழி செய்கின்றது. **Display Query** பொத்தானை தேர்வு செய்தால் உடனே வினவல் செயல்படுத்தப்படுகிறது. **Modify Query** பொத்தான் வினவலை மாற்றி அமைக்க வழி செய்கிறது. அங்கே வரிசை முறை, மற்றும் பெயர் ஆகியவற்றை மாற்றி அமைத்துக் கொள்ளலாம்.

ஒரு அட்டவணைக்கு எத்தனை வினவல்கள் வேண்டுமானலும் உருவாக்கலாம். வினவல்கள் உருவாக்கப்பட்டு சேமிக்கப்பட்டுவிட்டால் அவற்றை எப்பொழுது வேண்டுமானாலும் செயற்படுத்திக் கொள்ளலாம். அப்படிச் செய்ய வினாச் சாளரத்தில் தோன்றும் எந்த வினாக் குறியை வேண்டுமானாலும் இருமுறை 'கிளிக்' செயல்படுத்தலாம்.

7.6.2.1 வடிகட்டிகள் (Filters)

வடிகட்டி **(Filter)** என்பது ஒருவகை வினவலாகும் **(Query)** இதுவும், கொடுக்கப்பட்ட நிபந்தனைகளை நிறைவு செய்யும் பதிவுகளைத் தேர்ந்தெடுத்துக் காட்ட பயன்படுகிறது. வடிகட்டிக்கும் வினவலுக்கும் உள்ள ஒரே வேறுபாடு என்னவென்றால் வினவல்களை **(Query)** சேமித்து வைத்து பிறகு வேண்டிய போது பயன்படுத்திக் கொள்ளலாம். ஆனால், வடிகட்டி **(Filter)** யில் இந்த வசதி இல்லை. **Star Base**–இல் இரண்டு வகை வடிகட்டிகள் உள்ளன: அவையாவன:

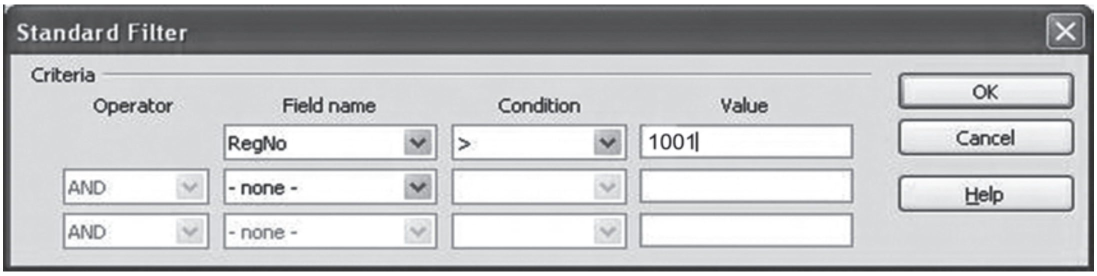
Auto Filter, Standard Filter என்பவையாகும்.

Auto Filter

கருவிப்பட்டை **(Toolbar)** யில் **Auto Filter** பணிக்குறி உள்ளது. இந்த பணிக்குறியை 'கிளிக்' செய்தால், அப்போதைக்குத் தேர்வு செய்த புலத்தில் என்ன தரவு **(Disk)** இருக்கின்றதோ அந்தத் தரவிற்குச் சரி சமமான தரவு எந்தெந்த பதிவுகளில் உள்ளதோ அந்த பதிவுகளை மட்டும் தேர்வு செய்து காட்டும்.

Standard Filter

நிபந்தனையோடு பயன்படுத்தப்படும் வடிகட்டிக்கு **Standard Filter (Default Filter)** என்று பெயர். கருவிப்பட்டையில் உள்ள **Standard Filter** ‘கிளிக்’ செய்து நிபந்தனையைக் கொடுக்க வேண்டும்.

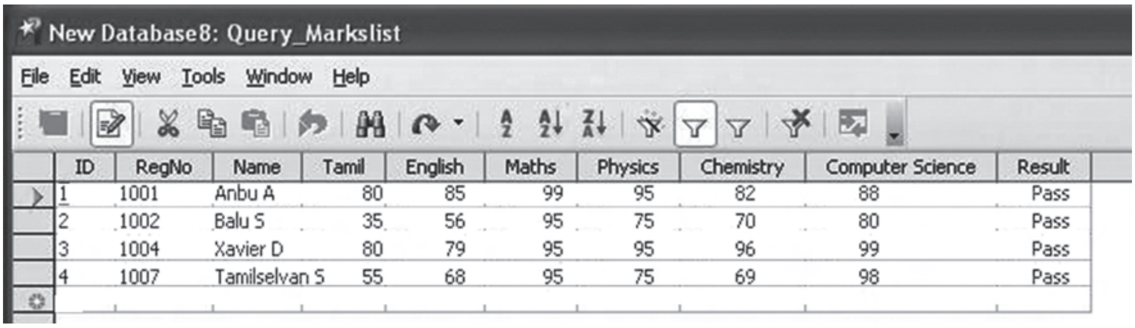


படம் 7.19 வடிகட்டி சன்னல் திரை

Filter சன்னல் திரை (படம் 7.19), நிபந்தனை கொடுக்கப் பயன்படுத்திய வினவல் **(Query)** திரை போன்றே இருக்கும். வடிகட்டிகளை நன்கு புரிந்து கொள்ள கீழ்க்கண்ட எடுத்துக்காட்டைப் பார்க்கவும். கணித பாடத்தில் 90 க்கு மேல் மதிப்பெண் பெற்ற மாணவர்களை பட்டியலிட விரும்புவதாகக் கொள்வோம். இதனைச் செய்ய கீழ்க்கண்ட வழிமுறையைப் பின்பற்றவும்.

- ஏற்கனவே **Query** சாளரத்தில் உள்ள வினவலை இருமுறை ‘கிளிக்’ செய்து திறக்கவும். எடுத்துக்காட்டாக**Query - Mark list**
- **Standard Filter** பொத்தானை ‘கிளிக்’ செய்க ஒரு **Filter** சன்னல் திரை தோன்றும்.
- **Field Name** பெட்டியிலிருந்து **‘Maths’** என்ற பாடப் பெயரைத்தேர்வு செய்யவும்.
- **Condition** பெட்டியிலிருந்து **>** என்ற குறியைத் தேர்வு செய்யவும்.
- **Value** பெட்டியில் 90 என்ற எண்ணைக் கொடுக்கவும்.
- **OK** பொத்தானை ‘கிளிக்’ செய்யவும்.

படம் 7.20 இல் உள்ளதைப் போல வடிகட்டி எடுக்கப்பட்ட பதிவுகள் காட்டப்படும். வடிக்கட்டியை நீக்க **Remove Filter / Sort** பணிக்குறியை ‘கிளிக்’ செய்க. இப்பொழுது மூல அட்டவணை எல்லா பதிவுகளுடனும் திரையில் தோன்றும்.



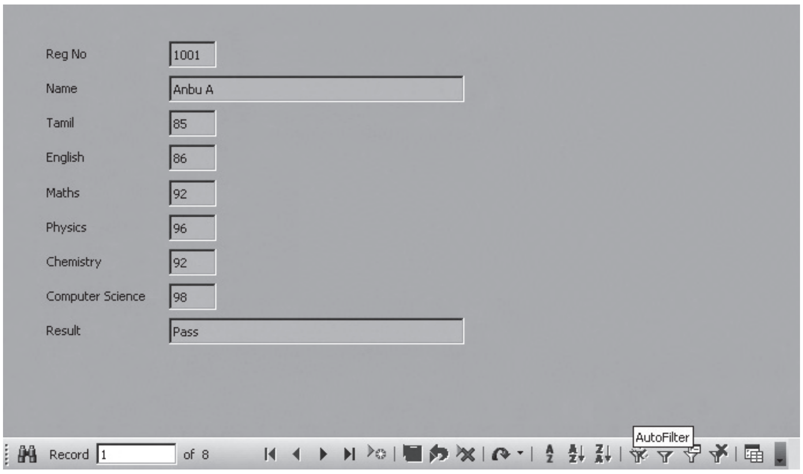
ID	RegNo	Name	Tamil	English	Maths	Physics	Chemistry	Computer Science	Result
1	1001	Anbu A	80	85	99	95	82	88	Pass
2	1002	Balu S	35	56	95	75	70	80	Pass
3	1004	Xavier D	80	79	95	95	96	99	Pass
4	1007	Tamilselvan S	55	68	95	75	69	98	Pass

படம் 7.20 வடிகட்டி எடுக்கப்பட்ட பதிவுகள்

இந்த பணிக்குறியை ‘கிளிக்’ செய்து வடிகட்டியை நீக்கலாம்.

7.6.3 படிவம் வடிவமைத்தல் (Form Designing)

தரவுகளை அட்டவணையில் உள்ளதைப்போல வரிசைகளிலும் நெடுவரிசைகளிலும் பயன்படுத்துகின்ற முறை, பதிவுகளை பார்க்கவும். சிறு பதிப்பாய்வு செயல்களை செய்வதற்கும் வசதியாக இருந்தாலும், இந்த முறை தரவுத்தள மேலாண்மை செயற்பாட்டிற்கு ஏற்றதல்ல. எடுத்துக்காட்டாக, ஒரு அட்டவணையில் ஏராளமான புலங்கள் இருக்கின்ற பொழுது அவற்றை ஒரே நேரத்தில் ஒரு சன்னல் திரையில் காட்ட முடியாது. இப்படிப்பட்ட அட்டவணைகளில் புலங்களைப் பார்க்க உருள்பட்டைகள் **(Scroll Bars)** அமைக்கப்படுகின்றன. ஆனால் உருள் பட்டை வடிவ அமைப்புகளில் ஏராளமான தரவுகளை கொடுப்பதில் மிகுந்த சங்கடங்கள் உண்டாகும். இப்படிப்பட்ட சூழ்நிலைகளை எளிதாக , பயன்பாட்டிற்கு ஏற்ற வகையில் படிவங்களை உருவமைத்து செயல்படுத்தலாம். **Star Base**–இல், நாம் பயணச்சீட்டுகள் அல்லது கல்லூரியில் சேருவதற்கான நிரப்பும் விண்ணப்பங்கள் போன்ற படிவங்கள் **(Forms)** உள்ளன. ஒரு படிவம் **(Form)** என்பது ஒரு பதிவின் புலங்களை நல்ல இடைவெளிகளோடு காட்டும் திரையாகும். ஒரு படிவத்தின் வடிவத்தை நாம் விரும்பியபடி பயன்பாட்டிற்கு ஏற்ப அமைத்துக் கொள்ளலாம். ஒரு படிவம், ஒன்று அல்லது அதற்கு மேற்பட்ட அட்டவணைகளோடு தொடர்பு கொண்டிருக்கலாம். படம் 7.21 –இல் **Mark list** – ஐத் தழுவிய ஒரு மாதிரிப்படிவம் காட்டப்பட்டுள்ளது.



படம் 7.21 Mark list தழுவிய படிவம்

Autopilot வசதியை பயன்படுத்தி படிவத்தை உருவமைத்தல், **Autopilot** வசதியை பயன்படுத்தி ஒரு வினவலை உருவாக்குவதை போன்றதாகும். **StarBase**–இல் படிவத்தை வடிவமைக்க கீழ்க்கண்ட வழிமுறையை பின்பற்றவும்:

1. **Database** சன்னல் திரையில் உள்ள **Form** என்ற பட்டியை தேர்வு செய். பிறகு **'Use Wizard to Create Form'** என்ற பட்டியைத் தேர்வு செய்யவும். படம் 7.15–இல் உள்ளதை போன்ற ஒரு சன்னல் திரை தோன்றும். இந்த திரையில் ஏற்கனவே உருவாக்கி சேமித்து வைக்கப்பட்டுள்ள அட்டவணைகள், வினவல்கள் இவற்றின் பட்டியல் இருக்கும்.

2. எந்த அட்டவணை அல்லது வினவலுக்கு படிவம் **(Form)** வடிவமைக்க விரும்புகின்றேமோ அதனை **Tables** அல்லது **Queries** பட்டியல் பெட்டியிலிருந்து தேர்வு செய்ய வேண்டும். இப்பொழுது தேர்வு செய்ய வேண்டும். இப்பொழுது தேர்வு செய்யப்பட்ட அட்டவணை அல்லது வினவலின் புலப்பெயர்கள் திரையில் **'Available Fields'** உரைப்பகுதியில் தோன்றும்.

3. **>, >>,** ஆகிய பொத்தான்களை பயன்படுத்தி எந்த புலத்தை படிவத்தில் சேர்க்க விரும்புகின்றோமோ அந்த புலப்பெயர்களைத் தேர்வு செய்யவும். பிறகு **Next** என்ற பொத்தானை 'கிளிக்' செய்யவும்.

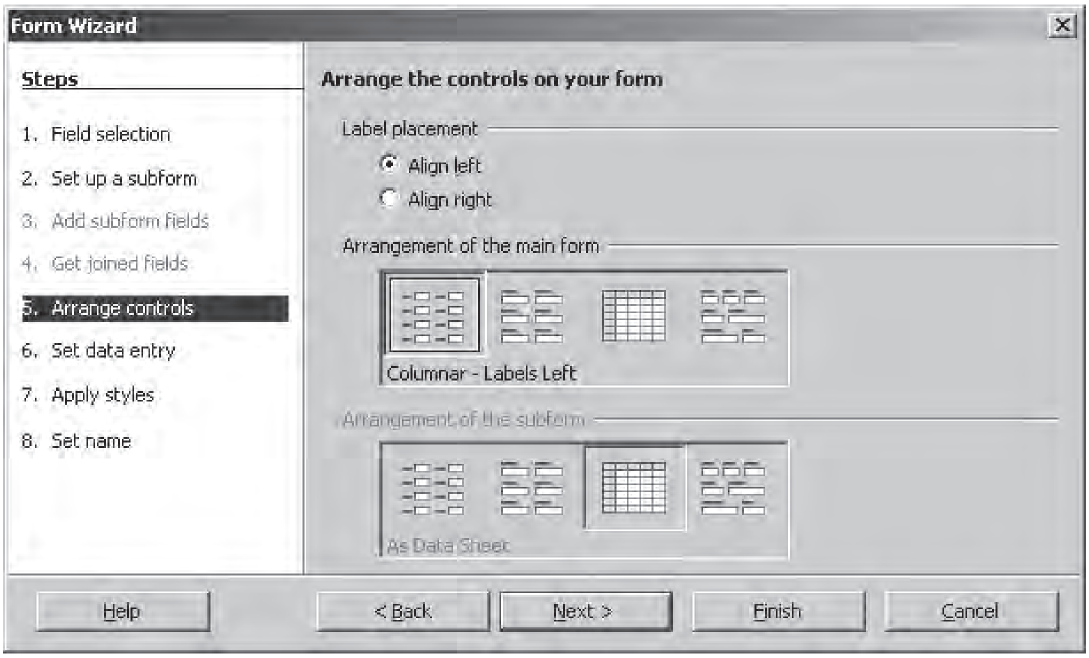
4. அடுத்து **Star office base 'Set up a subform'** சன்னல் திரையை படம் 7.22–இல் உள்ளதை போலக்காட்டும். ஒரு துணைப் படிவம் என்பது இன்னொரு படிவத்துக்குள் அமைக்கப்படும் படிவமாகும். அட்டவணை அல்லது வினவலிலிருந்து தரவை ஒன்றுக்கு பல உறவுகள் உள்ள விவரத்தைக் காட்ட துணைப்படிவத்தைப் பயன்படுத்தலாம். துணைப் படிவத்தை பயன்படுத்த விரும்பினால், **'Add subform'** பட்டியைத் தேர்வு செய்க. இல்லையெனில் **Next** பொத்தானை 'கிளிக்' செய்க. பிறகு படம் 7.23–இல் உள்ள திரை தோன்றும். இங்கே புலங்களை படிவத்தில் அமைக்கும் பாணியைத் தேர்வு செய்து கொள்ளலாம். Style குறிகள் தேர்வு செய்யப்படும் படிவம் எப்படித் தோன்றும் என்பதைக் காட்டும். Style குறி மீதும் **Next** பொத்தான் மீது 'கிளிக்' செய்க.

5. அடுத்து சன்னல் திரை (படம் 7.23) தரவைக் கொடுக்கும் முறையைத் தேர்வு செய்ய பயன்படுகிறது. **Set data entry** பொத்தானை 'கிளிக்' செய்து படிவம் புது தரவைக் கொடுக்க மட்டும் பயன்படுத்தப்படும் அல்லது அனைத்துத் தரவுகளையும் திரையிலிடப் பயன்படுத்தப்படும் என்பதை முடிவு செய்க.

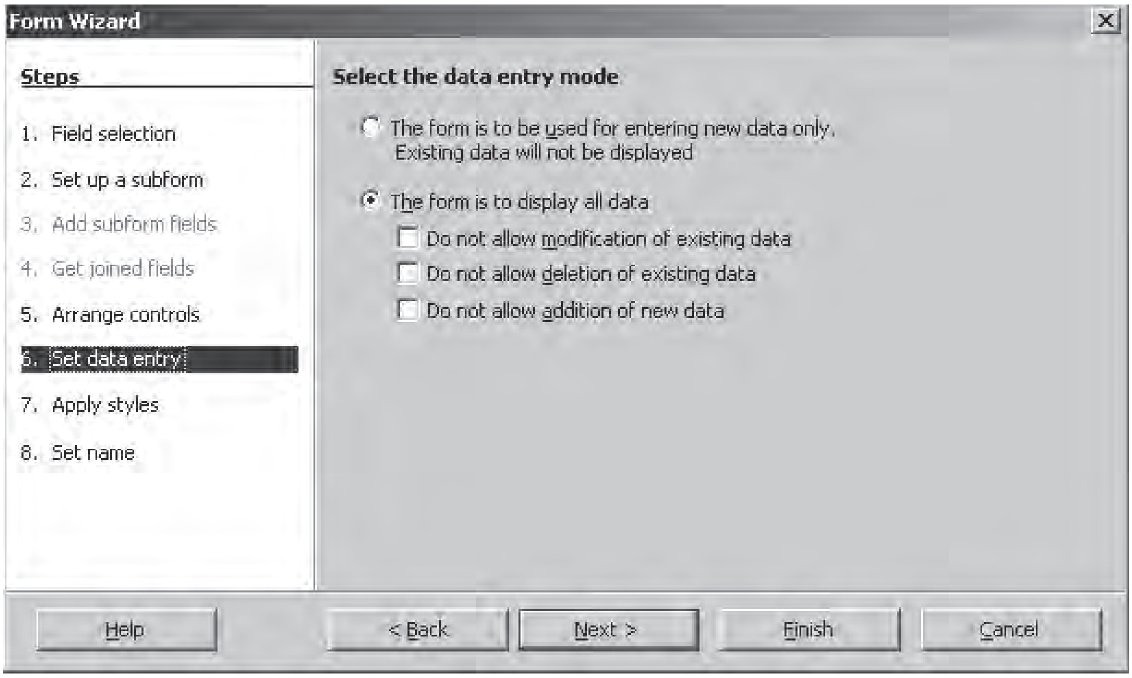
6. அடுத்த திரையில் உள்ள **Form wizard - Apply styles** படிவத்தின் பாணியை வரையறுக்கப் பயன்படுகிறது. பக்க அமைப்பின் பாணி, படிவத்தின் எழுத்துரு வகை, அளவு, நிறம் மற்றும் உரையின் பின்னணி ஆகியவற்றைத் தேர்வு செய்ய உதவுகிறது. **Field Border** படிவத்தின் புல எல்லைப் பாணியை வரையறுக்கப் பயன்படுகிறது. (படம் 7.24)

7. அடுத்து **set the name of the form** திரை (படம் 7.25) படிவத்திற்கு பெயர் கொடுக்கக் கோரித் தோன்றும், படிவத்தை சேமித்துத் தேக்கி வைக்கலாம். படிவ ஆவணமாகத் திறக்கலாம். திறந்து புது தரவைக் கொடுக்கலாம் அல்லது உள்ள தரவை திரையிடலாம். **Work with the form** படிவத்தை பதிப்பிக்க அணியமாக்குகிறது. **Modify the forms** படிவ அமைப்பை மாற்றி அமைக்கப் பயன்படுகிறது. படிவத்தின் பெயரை அச்சிட்டு **Finish** பொத்தானை அழுத்தவும்.

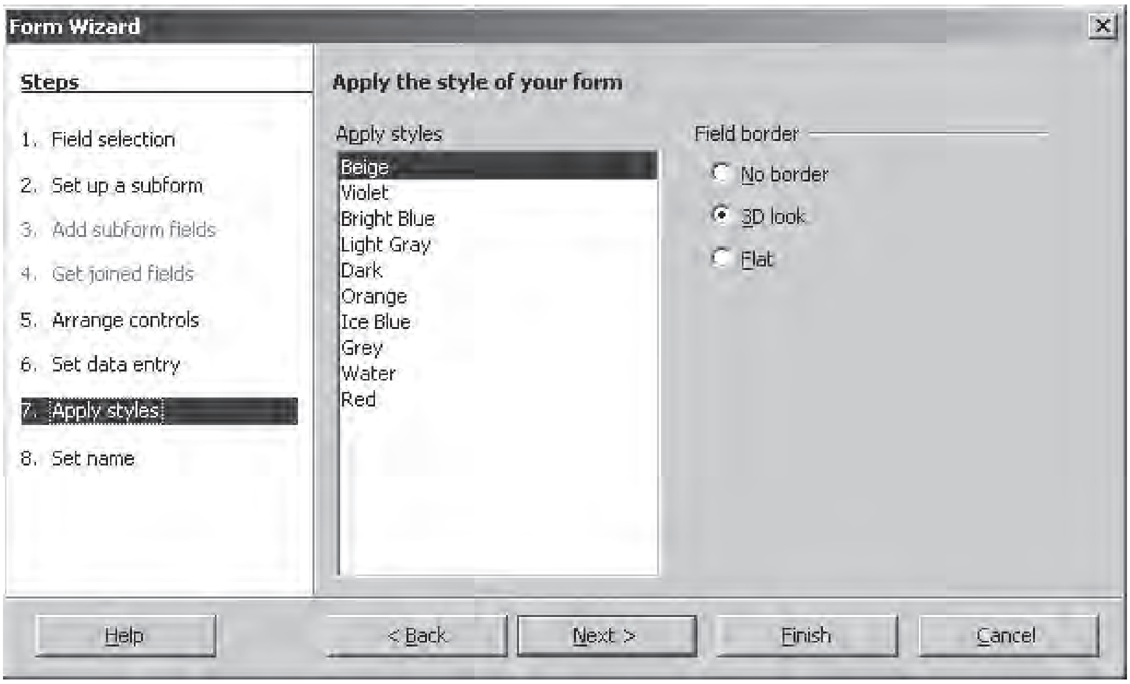
படிவத்தை உருவாக்கிய உடனே அதனை பயன்படுத்தி அட்டவணையில் உள்ள பதிவுகளை பார்க்கவோ, சேர்க்கவோ, நீக்கவோ, பதிப்பாய்வு செய்யவோ, செய்யலாம்.



படம் 7.22 Form Wizard சன்னல் திரை

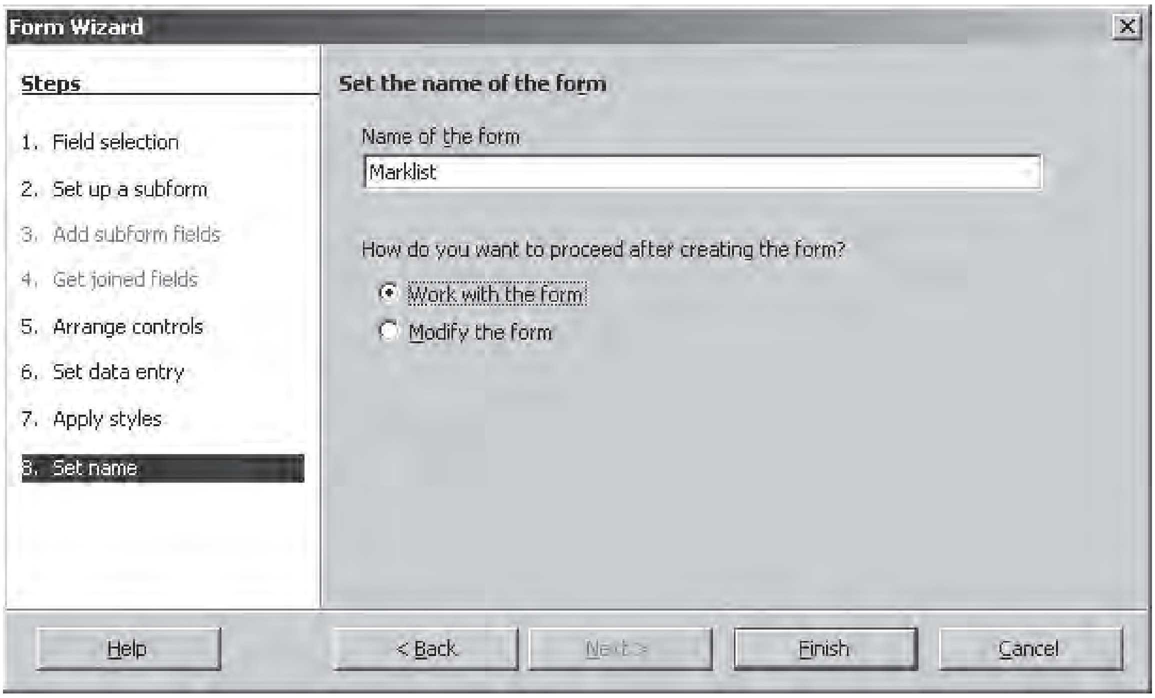


படம் 7. 23 **Data Entry Mode சன்னல் திரை**



படம் 7.24 **Styles சன்னல் திரை**

படிவம் ஒருமுறை உருவமைத்து விட்டால் அதனைப் பயன்படுத்தி, படிவத்தில் உள்ள பகுதிகளைப் பார்க்கலாம்; சேர்க்கலாம்; நீக்கலாம். ஒரு அட்டவணைஅல்லது வினவலுக்கு எத்தனை படிவங்கள் வேண்டுமானாலும் உருவமைத்துக் கொள்ளலாம். ஒரு முறை உருவமைத்து சேமித்த படிவத்தை **Database** சாளரத்தில் உள்ள **Forms** பொத்தானை கிளிக் செய்து பயன்படுத்தலாம். **Forms** சாளரத்தில் உள்ள **Form Name**–ஐ தேர்வு செய்து இருமுறை ‘கிளிக்’ செய்யவும்.



படம் 7.25 Setting Form Name சன்னல் திரை

செய்து பார்

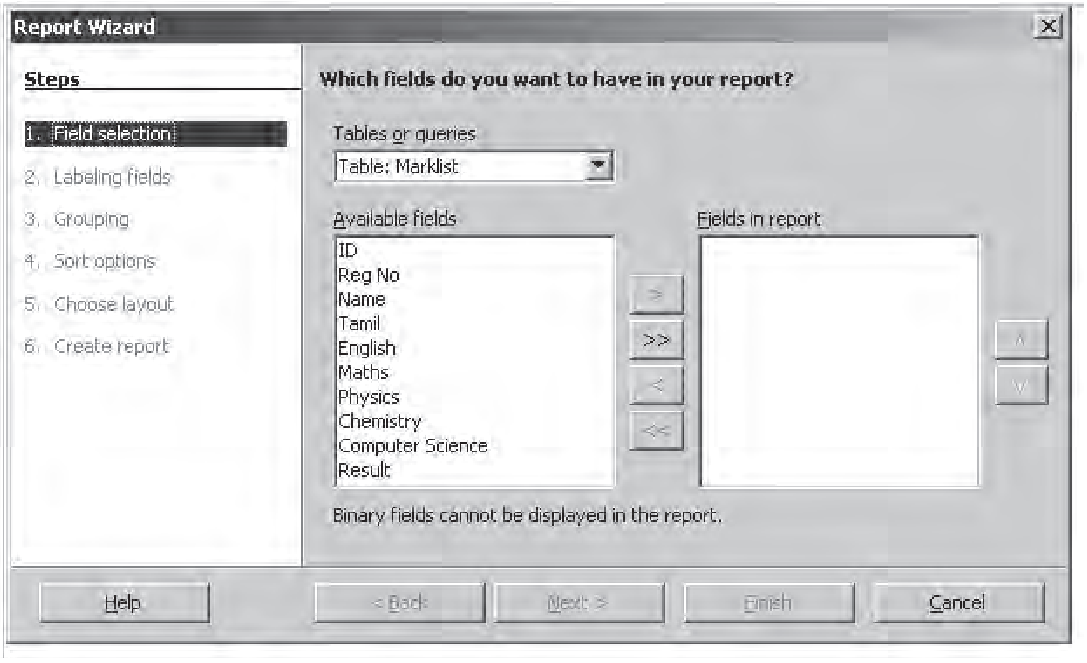
Mark list அட்டவணையை பயன்படுத்தி மாணவர்கள் அறிவியல் பாடங்களில் **Maths, Physics , Chemistry** ஆகிய பாடங்களில் எடுத்துள்ள மதிப்பெண்களை மட்டும் பார்ப்பதற்கான படிவம் ஒன்றை வடிவமைக்கவும்.

7.6.4 அறிக்கைகளை உருவாக்கல் (Generating Reports)

பயனாளரின் தேவைக்கேற்ப தரவுத்தளத்திலிருந்து தரவுகளைக் கொண்டதொகுக்கப்பட்ட தகவல்களை அச்சிட்ட வடிவில் வழங்குவதையே அறிக்கை **(Report)** என்கிறோம். அறிக்கைகள், எளிமையான பதிவுகளின் பட்டியல் முதல் கொண்டு பயனர்களின் பயன்பாட்டிற்கேற்ப உருவாக்கப்பட்ட வடிவமைப்புகள் வரை பலவகைப்பட்டதாக இருக்கலாம் அறிக்கைகளை உருவாக்கும் மென்பொருள்கள், தேர்வு செய்யப்பட்ட தரவுகளையும் கொடுக்கப்பட்ட தானியங்கு கணிப்பீட்டு முறைகளையும் கையாண்டு அறிக்கைகளைத் தானே உருவாக்கி அச்சிட்டு கொடுக்கின்றன.

பெரும்பாலான **DBMS** – தரவுத்தள மேலாண்மை அமைப்பு மென்பொருள்கள் வினவல்கள் **(Queries)** மூலம் அறிக்கைகளை உருவாக்குகின்றன.

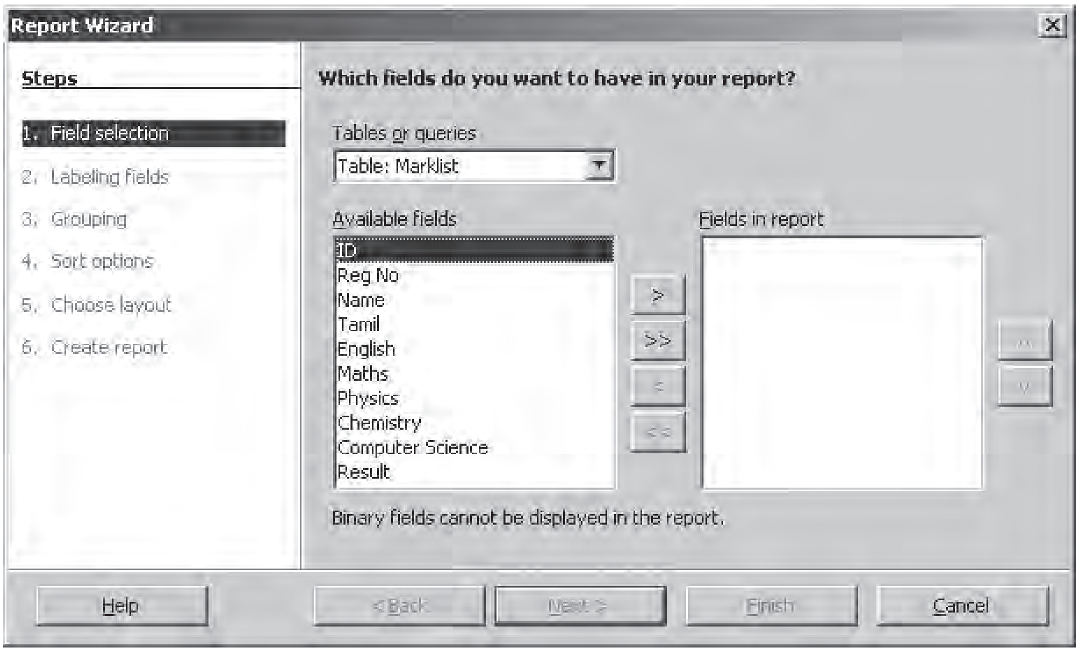
Star Base அறிக்கையை உருவாக்குவது ஒரு வினவலையோ படிவத்தையோ உருவாக்குவது போன்றதாகும். ஒரு அறிக்கையை உருவாக்க, **Datebase** சாளரத்தில் **(Pane)** உள்ள **Report** என்ற பட்டியை கிளிக் செய்யவும். பிறகு, **'Use wizard to Create Report'** என்ற பட்டிகளைத் தேர்வு செய்து 'கிளிக்' செய்யவும். இப்பொழுது படம் 7.25–இல் உள்ளதைப் போல **Report wizard** சன்னல் திரை தோன்றும்.



படம் 7. 26 Autopilot அறிக்கைகள்: அட்டவணைத் தேர்வு சன்னல் திரை

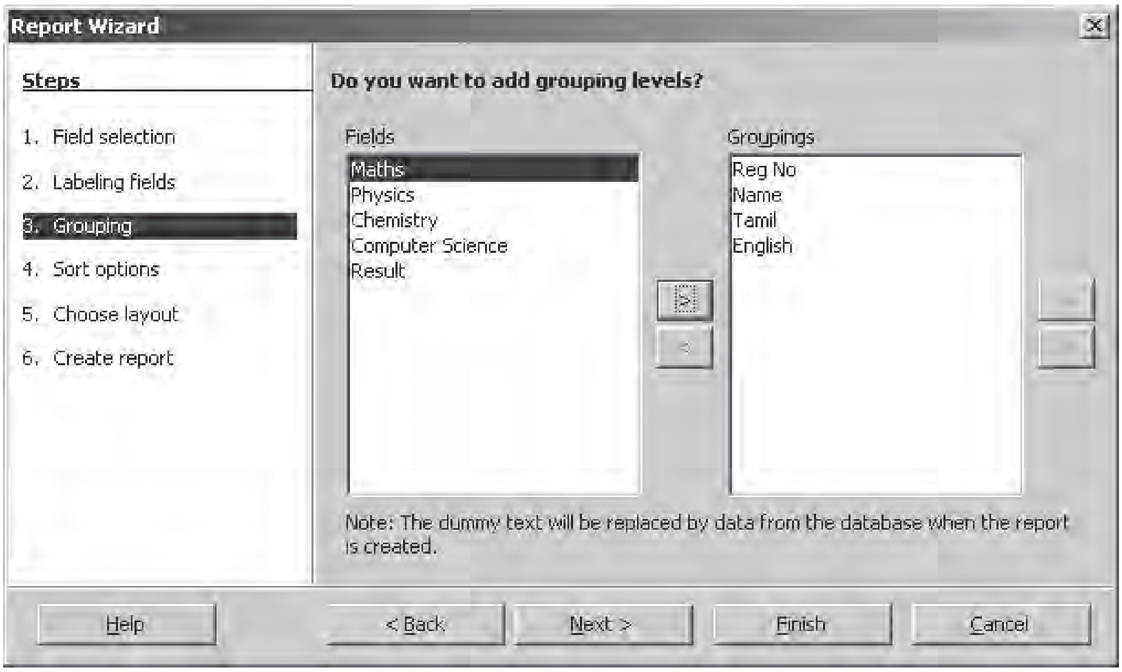
படிவத்தைப் போல இந்த சன்னல் திரை, ஏற்கனவே உருவாக்கி சேமிக்கப்பட்டுள்ள அட்டவணைகள் வினவல்கள் இவற்றை பட்டியலிட்டு காட்டுகிறது.

தேவையான அட்டவணை அல்லது வினவலைத் தேர்வு செய். படம் 7.26–இல் உள்ளதை போல திரையில் தேர்வு செய்யப்பட்ட அட்டவணை அல்லது வினவலின் புலங்களின் பட்டியலைக் கொண்ட சன்னல் திரை ஒன்று தோன்றும். இதிலிருந்து **>, >>** என்ற பொத்தான்களை பயன்படுத்தி அறிக்கையில் சேர்க்க வேண்டிய புலங்களை தேர்வு செய்யவும். பிறகு **Next** என்ற பொத்தானை ‘கிளிக்’ செய்யவும்.



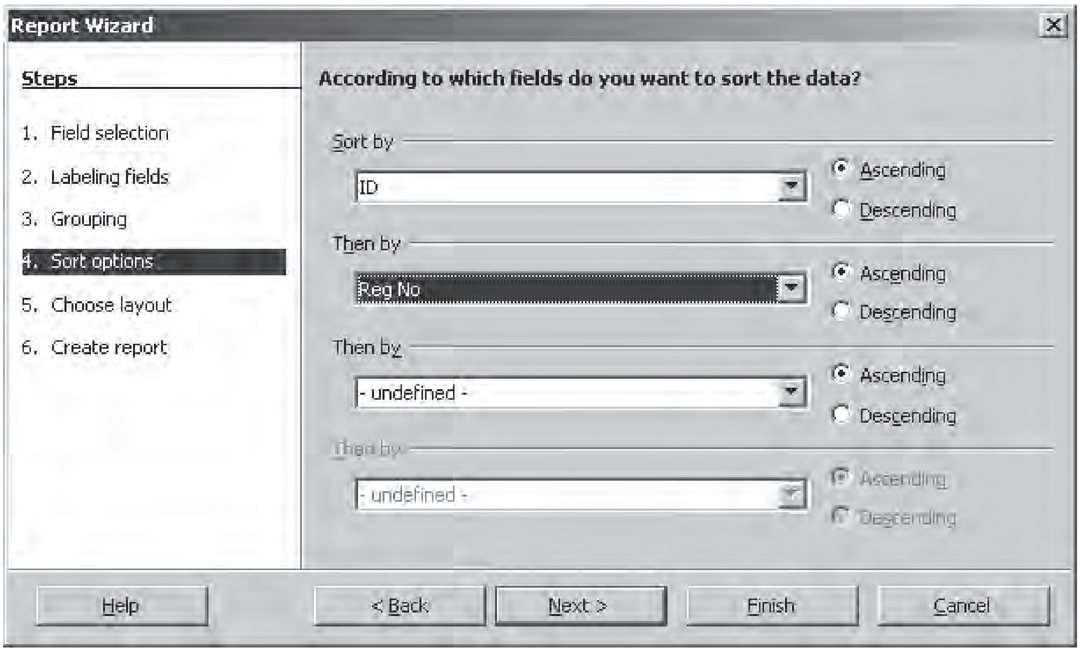
படம் 7.27 Field Selection சன்னல் திரை

இப்பொழுது **Labeling fields** சன்னல் திரை தோன்றும் (படம் 7.27). தேவையானல் புலத்தையும், அடையாளத்தையும் **(Label)** மாற்றி அமைக்கலாம். அடுத்து **Next** பொத்தானை 'கிளிக்' செய்யவும். இப்பொழுது **Sort Options** சன்னல் திரை (படம் 7.28) தோன்றும் எந்தெந்த புலங்களின் அடிப்படையில் பதிவு ஒன்றாக தொகுக்கப்பட வேண்டுமோ அவற்றை குறிக்க இந்த சன்னல் திரை பயன்படுகிறது.



படம் 7.28 Out - Line சன்னல் திரை

குறிப்பு : தொகுப்புப் புலத்தை வரையறுக்க வேண்டிய தேவையில்லை.

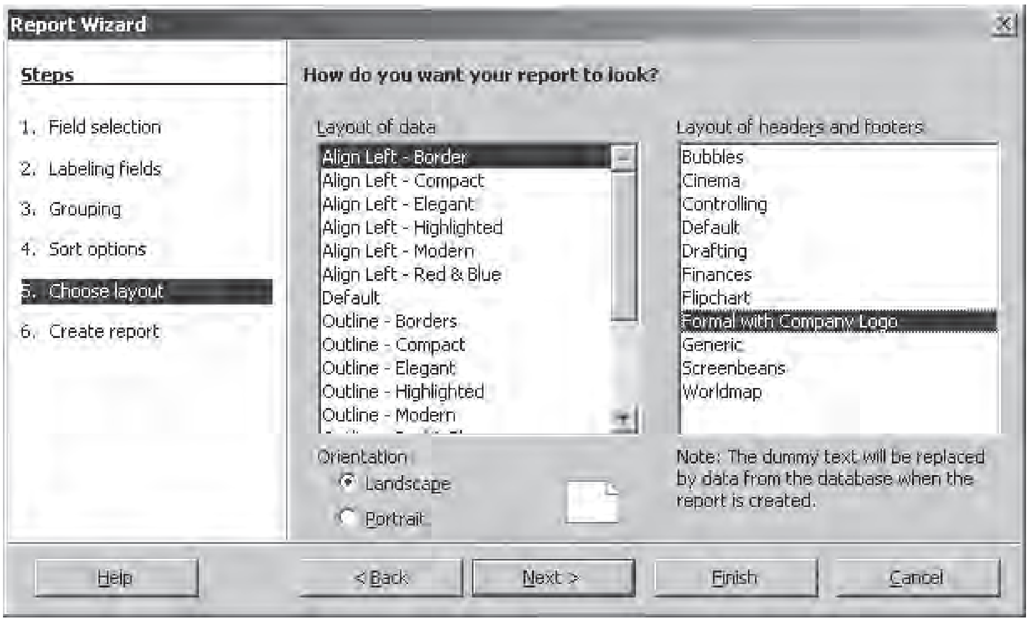


படம் 7.29 Sort Options சன்னல் திரை

Sort option - ஐத் தொடர்ந்து **Grouping** சன்னல் திரை (படம் 7.29) தோன்றும். இங்கே, **Layout of data, Layout of Headers & Footers and Orientation** ஆகியவற்றைத் தேர்வு செய்து அறிக்கை நமது தேவைக்கு ஏற்றவாறு அமைத்துக் கொள்ளலாம்.

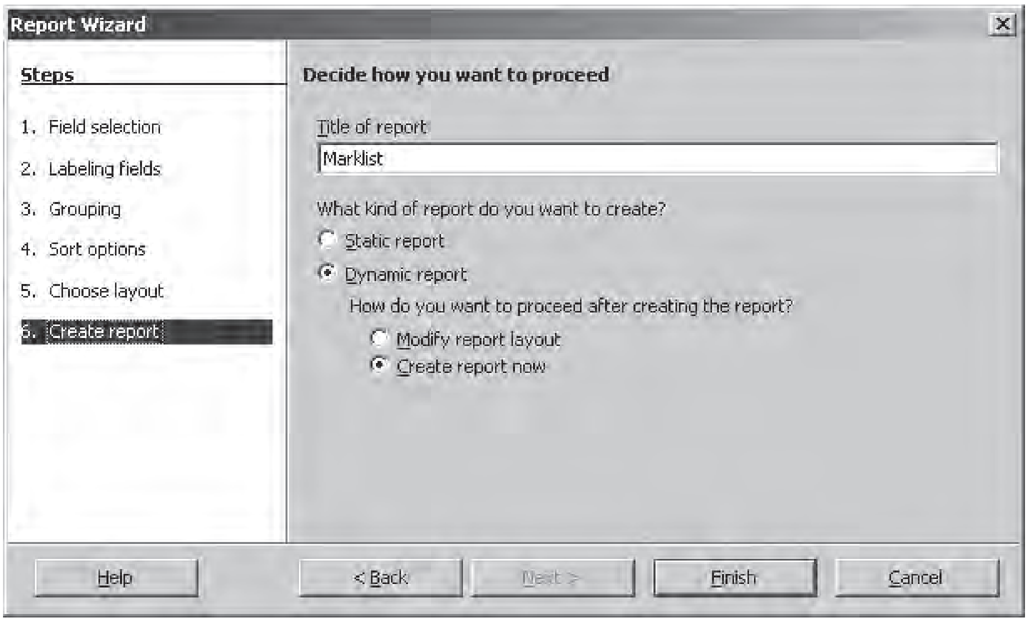
அடுத்து **Next** பொத்தானை 'கிளிக்' செய்த உடன் **Report wizard** சன்னல் திரை தோன்றும். (படம் 7.31)

இது நிலையான அல்லது மாற்றவல்ல அறிக்கையைத் தேர்வு செய்ய வழி செய்கிறது.



படம் 7.30 Choose Layout சன்னல் திரை

அறிக்கையை உடனடியாக பயன்படுத்தவோ மாற்றி அமைக்கவோ செய்யலாம். தலைப்பை அச்சிட்டு **Finish** பொத்தானை 'கிளிக்' செய்தால் அறிக்கையைப் பார்க்கலாம்.



படம் 7.31 Create சன்னல் திரை

ஒரு அட்டவணை அல்லது வினவலுக்கு பலவகை அறிக்கைகளை உருவமைத்து சேமித்து தேக்கி வைக்கலாம். சேமித்து வைத்த அறிக்கையை, சேமித்து வைத்த படிவம் அல்லது வினவலைப் பயன்படுத்துவது போலவே பயன்படுத்திக் கொள்ளலாம். தரவுத்தள சாளரத்தில் உள்ள **Reports**–ஐ கிளிக் செய்து, **Reports** சாளரத்தில் உள்ள **Name**–ஐ இருமுறை 'கிளிக்' செய்யவும்.

செய்து பார்

1. **Mark list** அட்டவணையில் உள்ள எல்லா பதிவுகளையும் காட்டும் அறிக்கை ஒன்றை உருவாக்கவும்.
2. **Mark list** அட்டவணையிலிருந்து தமிழ் பாடத்தில் தரநிலை **A** வும் அதற்கு மேலும் பெற்ற மாணவர்களைக் காட்டும் ஒரு அறிக்கையை உருவாக்கவும். (முதலில் ஒரு வினவலை உருவாக்கி பிறகு அறிக்கையை உருவாக்கவும்.)

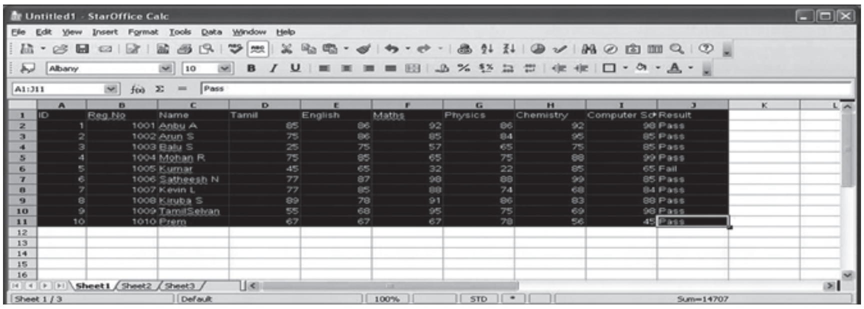
7.7 அலுவலகப் பணி தானியக்க பயன்பாடுகளுடன் ஒருங்கிணைத்தல். (Intergrating with office Automation Application)

Star Office பயன்பாடுகள் ஒன்றுக்கொன்று நெருக்கமாக பிணைக்கப்பட்டுள்ளன. எனவே ஒரு பயன்பாட்டிலிலுள்ள செய்திகளை இன்னொரு பயன்பாட்டில் செயல்படுத்துவது எளிது. அடுத்து வரும் பிரிவுகள் இந்த வசதியை எப்படி பயன்படுத்திக் கொள்ளலாம் என்பதை சில எடுத்துக்காட்டுகளுடன் விளக்குகின்றன.

7.7.1 ஒரு உரை ஆவணத்தில் Calc Cell பயன்பாட்டைச் செருகல்

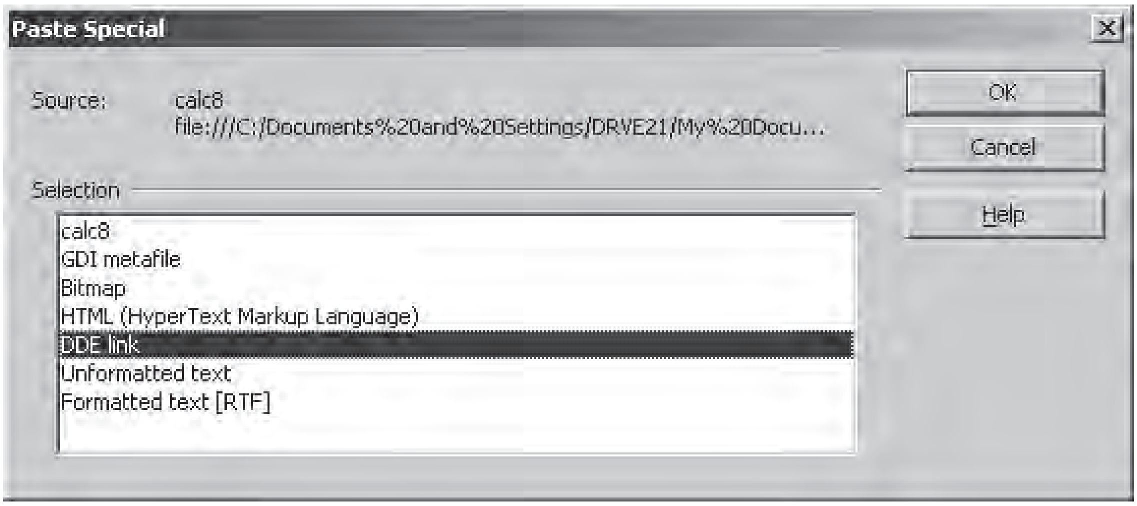
Star Writer பயன்பாட்டில் உள்ள ஒரு உரை ஆவணத்தில் Calc Cell வரிசை ஒன்றை செருகிவிட்டால், அட்டவணைச் செயலி (Spread Sheet) இல் உள்ள வாய்பாடுகளை மாற்றி அமைக்கும் போது, தரவுகள் புதிய வாய்ப்பாடுகளுக்கேற்ப தானே மாறிவிடும். இதனைச் செயல்படுத்த கீழ்க்குறிப்பிட்டுள்ளவாறு செய்க:

1. Star Office Writer (உரை) ஆவணத்தை திறக்கவும்.
2. தரவுகளைக் கொண்டுள்ள Star Office Calc (அட்டவணைச் செயலி)–ஐத் திறக்கவும்.
3. அட்டவணைச் செயலியிலிருந்து உரை ஆவணத்தில் அட்டவணையாகக் காட்ட விரும்பும் கண்ணி வரிசையைத் (Cell range) தேர்வு செய்க. தேர்வு செய்யப்பட்ட கண்ணி வரிசையை படம் 7.32–இல் காணலாம்.



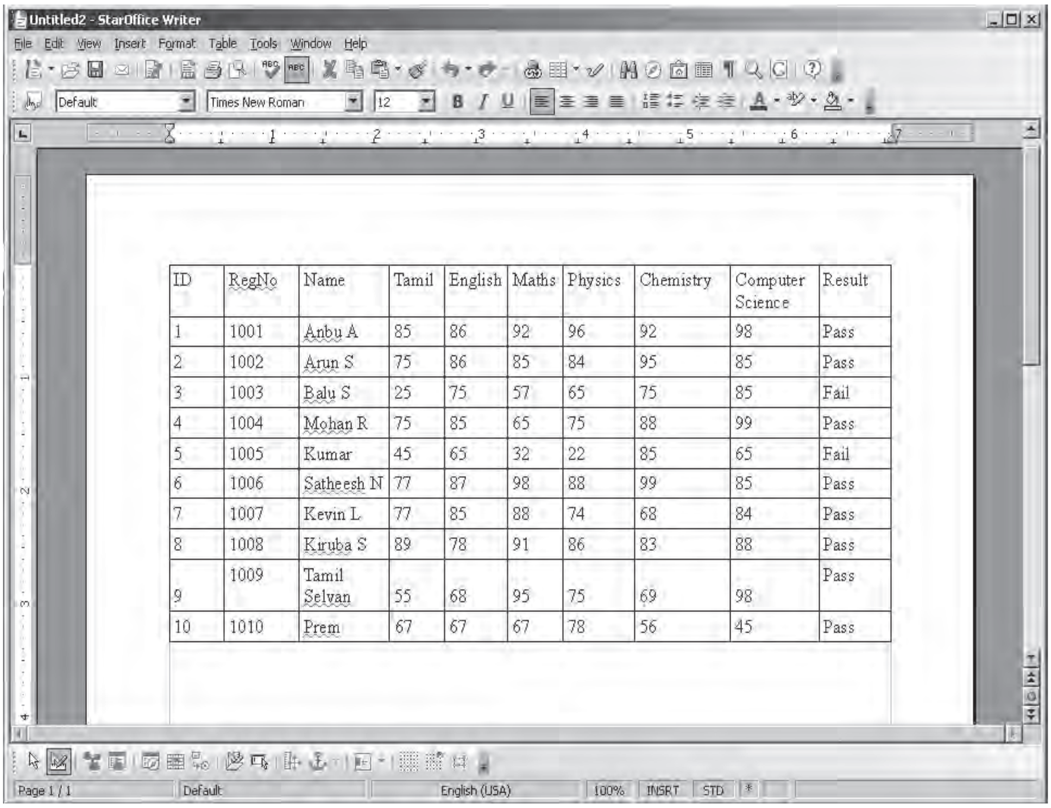
படம் 7.32 Selected Cell range in Spread Sheet

4. Edit →Copy இணைப்புகளைத் தேர்வு செய்க.
5. உரை ஆவணத்திலிருந்து Edit → Paste Special இணைப்புகளைத் தேர்வு செய்க.
6. Paste Special உரையாடல் பெட்டியிலிருந்து DDE link என்ற இணைப்பைத் தேர்வு செய்து கிளிக் செய்யவும். (படம் 7.33).



படம் 7.33 Paste Special dialog box

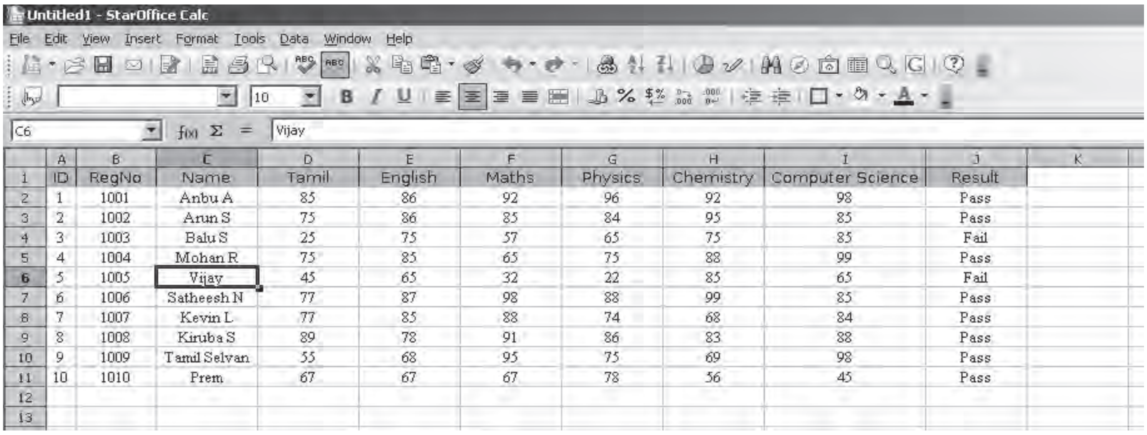
7. OK பொத்தானை கிளிக் செய்யவும். செருகப்பட்ட கண்ணி வரிசை உரை ஆவணத்தில் உள்ளதை படம் 7.34 –இல் காணலாம்.



ID	RegNo	Name	Tamil	English	Maths	Physics	Chemistry	Computer Science	Result
1	1001	Anbu A	85	86	92	96	92	98	Pass
2	1002	Arun S	75	86	85	84	95	85	Pass
3	1003	Balu S	25	75	57	65	75	85	Fail
4	1004	Mohan R	75	85	65	75	88	99	Pass
5	1005	Kumar	45	65	32	22	85	65	Fail
6	1006	Satheesh N	77	87	98	88	99	85	Pass
7	1007	Kevin L	77	85	88	74	68	84	Pass
8	1008	Kiruba S	89	78	91	86	83	88	Pass
9	1009	Tamil Selvan	55	68	95	75	69	98	Pass
10	1010	Prem	67	67	67	78	56	45	Pass

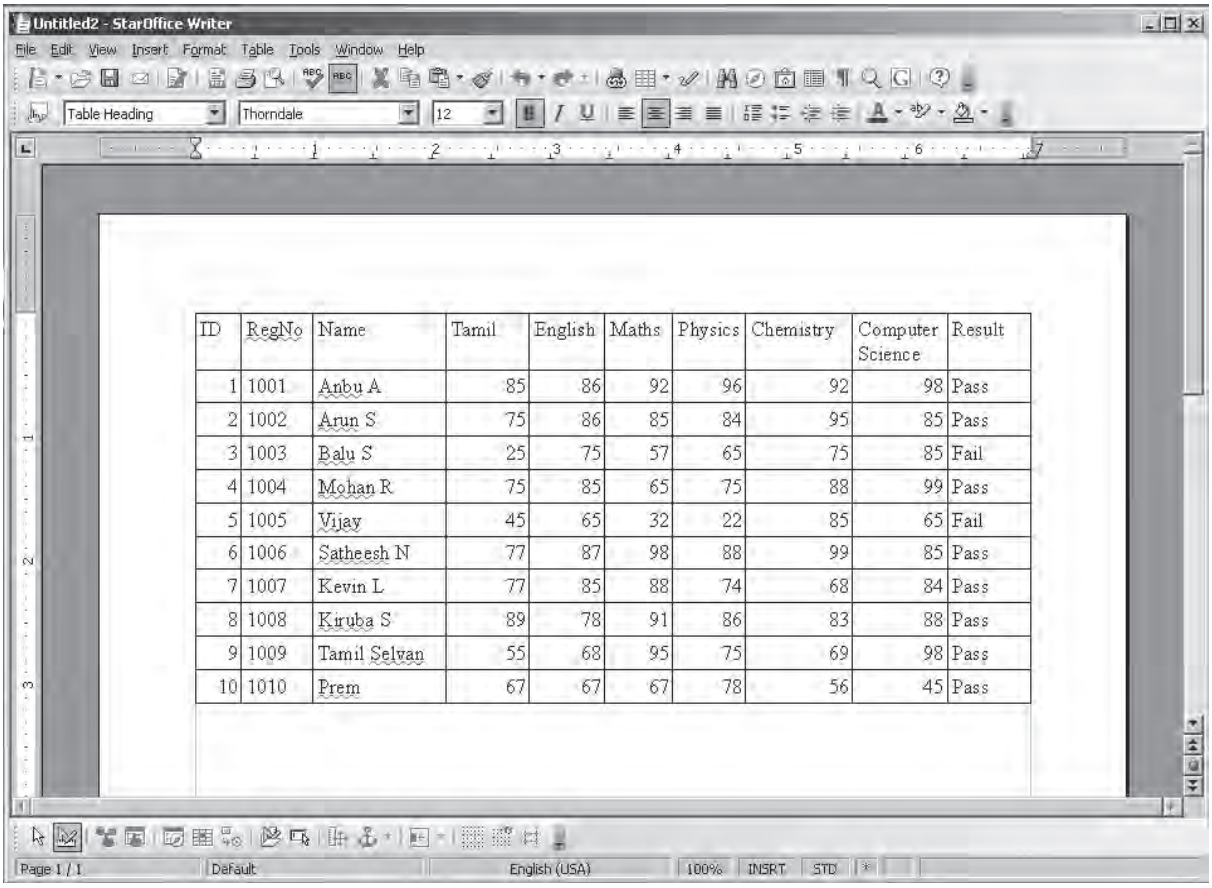
படம் 7.34 Text document with data copied from Spread Sheet

8. இப்பொழுது அட்டவணைச் செயலியில் உள்ள தரவை மாற்றிப் பார்க்கவும். உரை ஆவணத்தில் தரவுகள் தானே மாறுவதைப் பார்க்கலாம்.



ID	RegNo	Name	Tamil	English	Maths	Physics	Chemistry	Computer Science	Result
1	1001	Anbu A	85	86	92	96	92	98	Pass
2	1002	Arun S	75	86	85	84	95	85	Pass
3	1003	Balu S	25	75	57	65	75	85	Fail
4	1004	Mohan R	75	85	65	75	88	99	Pass
5	1005	Vijay	45	65	32	22	85	65	Fail
6	1006	Satheesh N	77	87	98	88	99	85	Pass
7	1007	Kevin L	77	85	88	74	68	84	Pass
8	1008	Kiruba S	89	78	91	86	83	88	Pass
9	1009	Tamil Selvan	55	68	95	75	69	98	Pass
10	1010	Prem	67	67	67	78	56	45	Pass

படம் 7.35 Data Modified in Spread Sheet

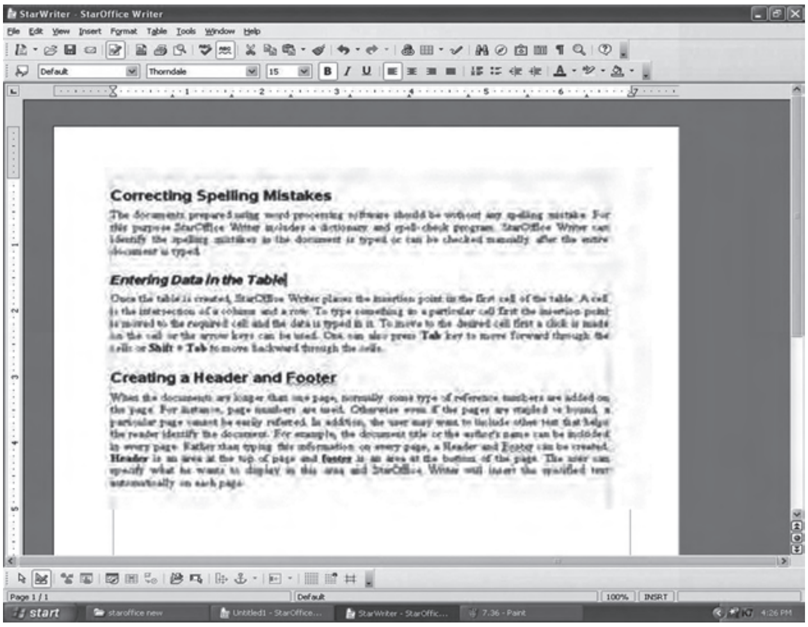


ID	RegNo	Name	Tamil	English	Maths	Physics	Chemistry	Computer Science	Result
1	1001	Anbu A	85	86	92	96	92	98	Pass
2	1002	Arun S	75	86	85	84	95	85	Pass
3	1003	Balu S	25	75	57	65	75	85	Fail
4	1004	Mohan R	75	85	65	75	88	99	Pass
5	1005	Vijay	45	65	32	22	85	65	Fail
6	1006	Satheesh N	77	87	98	88	99	85	Pass
7	1007	Kevin L	77	85	88	74	68	84	Pass
8	1008	Kiruba S	89	78	91	86	83	88	Pass
9	1009	Tamil Selvan	55	68	95	75	69	98	Pass
10	1010	Prem	67	67	67	78	56	45	Pass

படம் 7.36 Data Automatically updates in text document

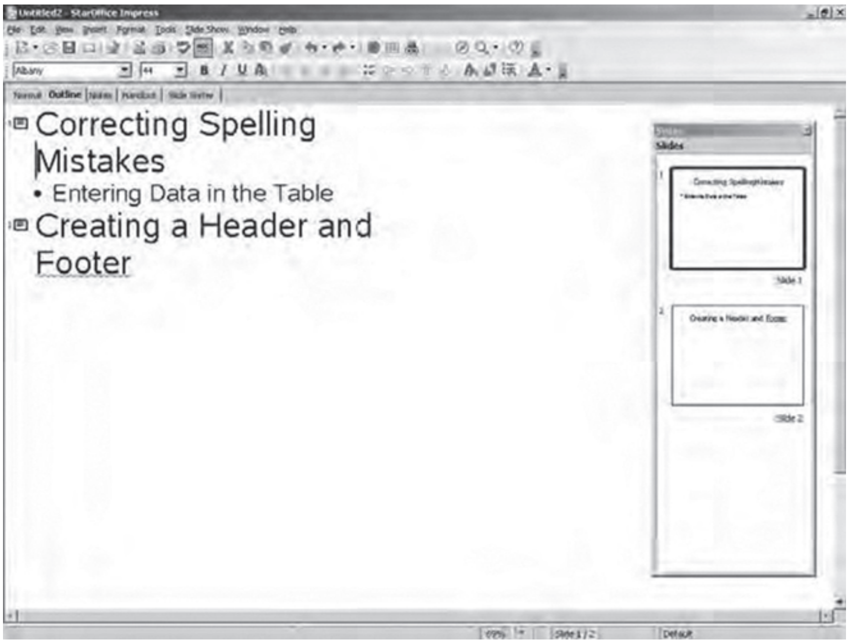
7.7.2 நிகழ்த்தலுக்குள் உரை வரிசைகளைச் செருகல்

உரை வரிசை என்பது பத்திகளுக்குக் கொடுக்க விரும்பும் " தலைப்பு–1 ", "தலைப்பு –2 " …. போன்ற தலைப்புகளைக் கொண்ட ஒரு தொகுப்பு ஆகும். படம் 7.37 உரைத் தலைப்புகளைக் காட்டுகிறது. உரை ஆவணத்தில் உள்ள தலைப்புகளை நிகழ்த்தலில் கொண்டு சேர்க்க கீழ்க்குறிப்பிட்டுள்ளவாறு செய்க.



படம் 7.37 **Text document with outline**

1. உரைத் தலைப்புகளைக் கொண்ட உரை ஆவணத்திலிருந்து **File → Send → Outline to Presentation** என்ற இணைப்புகளைத் தேர்வு செய்க.



படம் 7.38 **Presentation document with outline applied**

2. பிறகு மேலும் பல உரைத்தலைப்புகளை உரை வரிசையில் சேர்க்கலாம். இந்தச் செயலை Normal View பொத்தானை அழுத்தியும் செயற்படுத்தலாம்.

படம் 7.38 புதிதாக உருவமைக்கப்பட்ட நிகழ்த்தல் ஆவணத்தைக் காட்டுகிறது. ஒவ்வொரு தலைப்பும் ஒரு புது நிகழ்த்தல் சில்லுவைக் **(Slide)** காட்டுகிறது. சில்லுவில் தலைப்புகள் புள்ளிக் குறியோடு காட்டப்படுகின்றன.

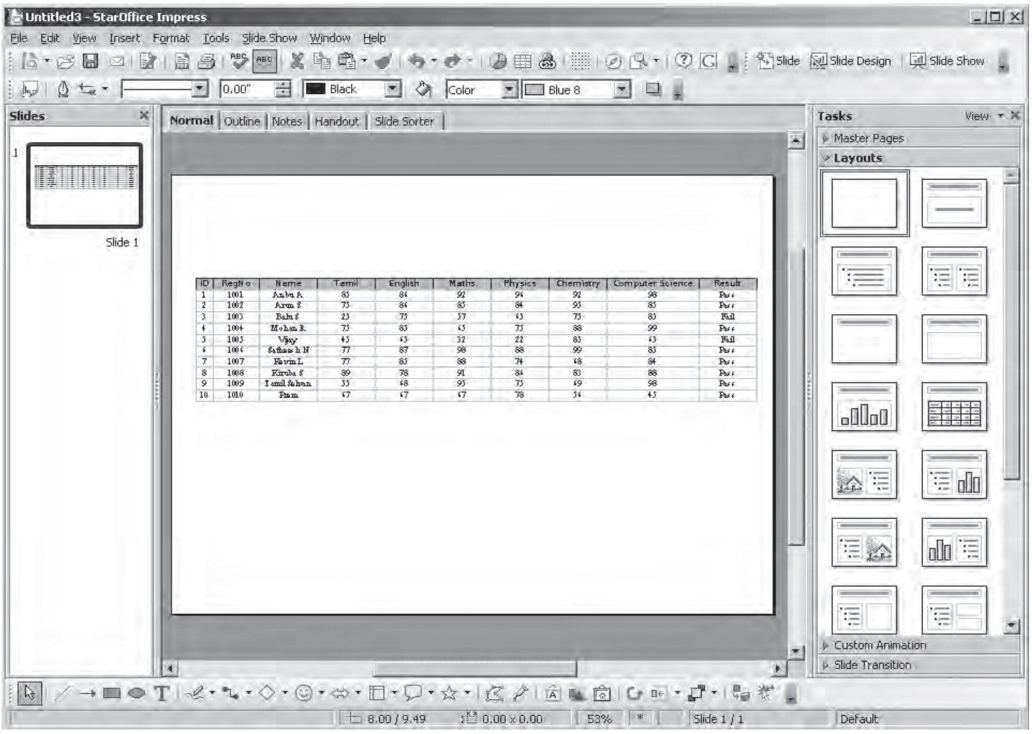
தலைப்புகளுடன் சேர்த்து பத்தியில் உள்ள உரைகளையும் அனுப்ப, **File→Send→Auto Abstract to Presentation Command** இணைப்புகளைத் தேர்வு செய்க.

7.7.3 இழுத்துச் சேர்த்துத் தரவுகளை நகலாக்கல்

Star Office பயன்பாடுகளுக்குள் ஒன்றிலிருந்து மற்றொன்றிற்கு உரை அல்லது தலைப்புகளை நகர்த்த அல்லது நகலாக்க, இழுத்துச் சேர்க்கும் **(Drag - and - Drop)** வசதியைப் பயன்படுத்தலாம். அட்டவணைச் செயலியின் கண்ணி வரிசையை நிகழ்த்தல் சில்லுக்குள் நகர்த்த கீழ்க்குறிப்பிட்டுள்ளவாறு செய்க:

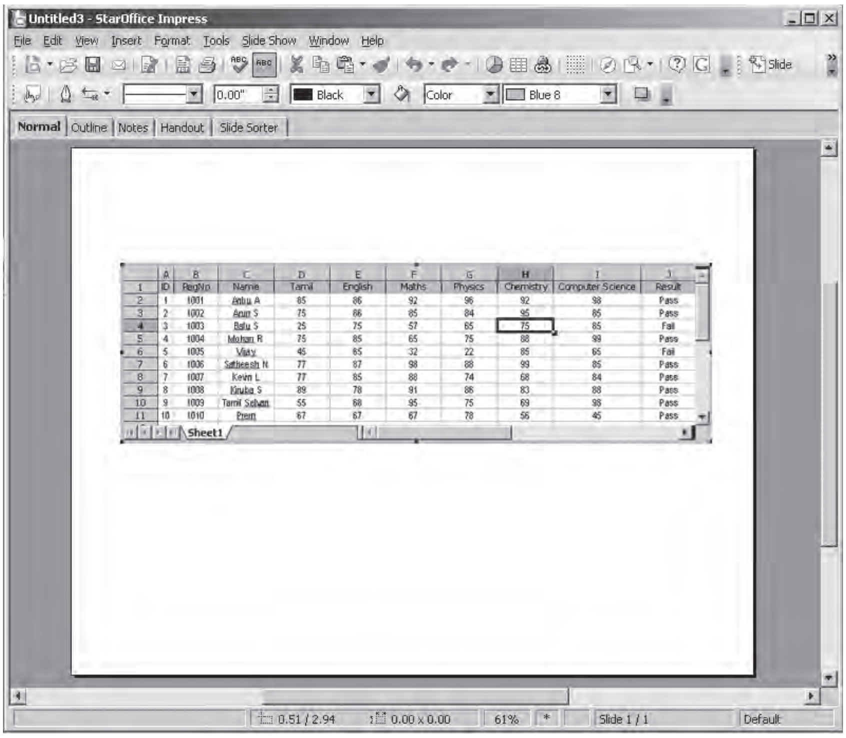
1. **Star Office Impress Presentation** பயன்பாட்டைத் திறக்கவும்.
2. நகல் செய்ய விரும்பும் தரவுகளைக் கொண்ட **Calc** அட்டவணைச் செயலியைத் திறக்கவும்.
3. அட்டவணைச் செயலியிலிருந்து நகல் செய்ய விரும்பும் கண்ணி வரிசையைத் தேர்வு செய்க.
4. **Alt + Tab** பொத்தான்களின் உதவியுடன் நிகழ்த்தல் சன்னலைத் திறந்து, இழுத்துச் சேர்த்தல் வசதியைப் பயன்படுத்தி தேர்வு செய்துள்ள கண்ணி வரிசையை இழுத்து நிகழ்த்தல் சில்லுவினுள் இடவும்.

இப்படி நகர்த்தி நகல் செய்யப்பட்ட கண்ணி வரிசை படம் 7.39–இல் காட்டப்பட்டுள்ளது.



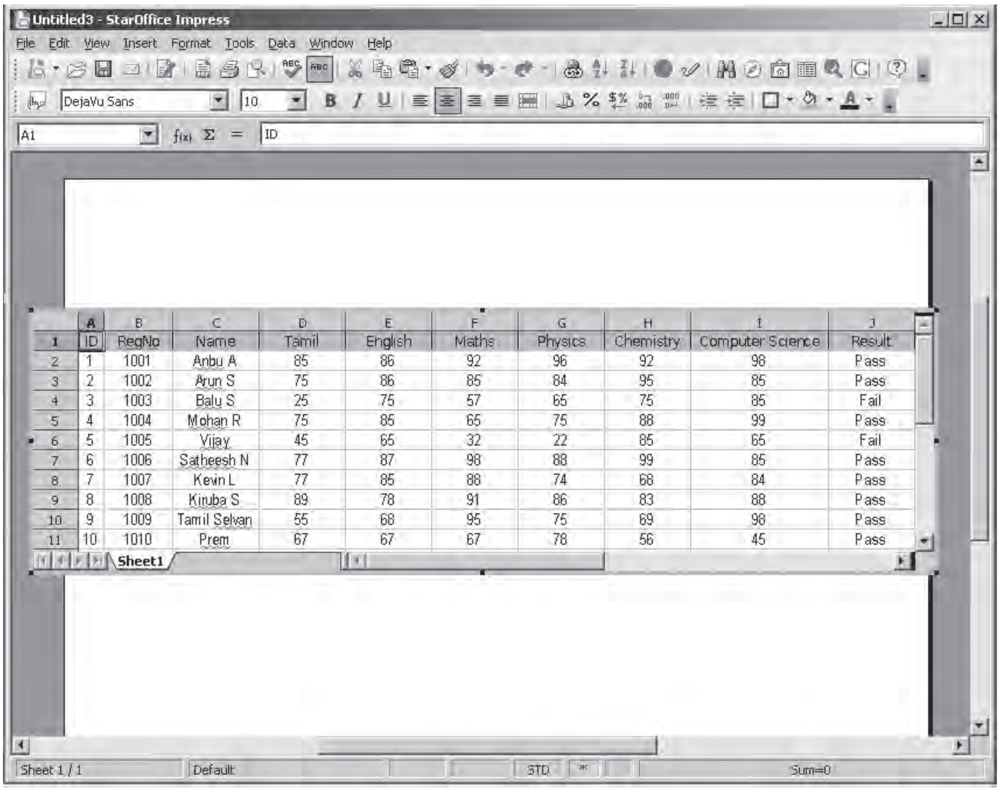
படம் 7.39 Cell range inserted as a plug -in in Presentation

நிகழ்த்தல் சில்லுவினுள் நகலாக்கிய கண்ணியில் உள்ள தரவுகளை மாற்றி அமைக்க விரும்பினால் கண்ணி மீது இருமுறை 'கிளிக்' செய்யவும். உடனே **Calc** பட்டிகளும் கருவிப் பட்டிகளும் தோன்றி கண்ணியை பதிப்பு நிலைக்குக் கொண்டு வந்து தரவுகளை மாற்ற வழி செய்யவும். பதிப்பு நிலையிலிருந்து மீள **Plug - in** பகுதிக்கு வெளியில் 'கிளிக்' செய்யவும். நிகழ்த்தல் சில்லுவில் பதிப்பு நிலையில் உள்ள கண்ணி படம் 7.40 –இல் காட்டப்பட்டுள்ளது.



படம் 7.40 **Edit Mode of the copied cell in the Presentation**

ஒரு கண்னி வரிசையை **Calc** அட்டவணைச் செயலியிலிருந்து **Writer** உரை ஆவணத்திற்குள் நகலாக்க மேற்கண்ட அதே முறையைப் பின்பற்ற வேண்டும். அவ்வாறு செயல்படுத்திய உரை ஆவணம் படம் **7.41**–இல் காட்டப்பட்டுள்ளது.



படம் 7.41 **Edit mode of the copied cell in the text document**

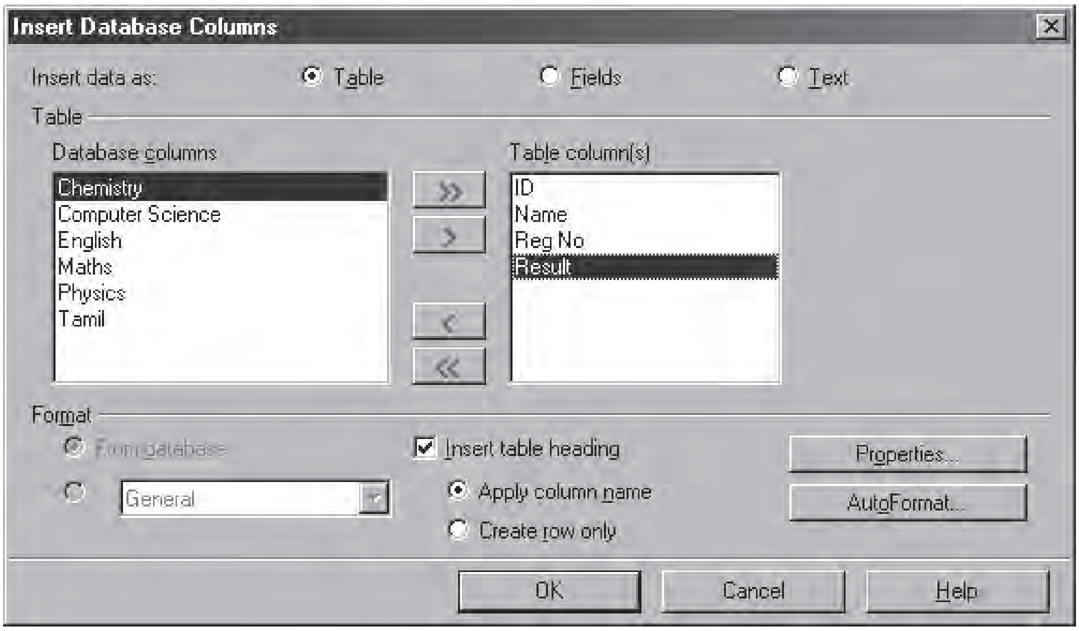
1.6.4 தரவுத்தள அட்டவணை ஒன்றை உரை ஆவணத்தில் சேர்த்தல்

தரவுத்தள அட்டவணை ஒன்றை உரை ஆவணத்தில் சேர்க்க கீழே குறிப்பிட்டுள்ளவாறு செய்க:

1. **Star Office Base** பயன்பாட்டைத் திறக்கவும். பிறகு தரவுத்தள பகுதியிலிருந்து அட்டவணைப் பகுதியையும், அட்டவணைப் பகுதியிலிருந்து வேண்டிய தரவுகளைக் கொண்ட அட்டவணையையும் தேர்வு செய்க.

2. பிறகு **Edit→Copy** இணைப்புகளைத் தேர்வு செய்க அல்லது **Ctrl + C** பொத்தான்களை 'கிளிக்' செய்க.

3. **Star Office Writer**–ஐத் திறந்து **Edit →Paste** இணைப்புகளைத் தேர்வு செய்க.



படம் 7.42 Inserted Database columns dialog box

4. தரவு அட்டவணையாக சேர்க்கப்பட வேண்டுமா அல்லது உரையாகச் சேர்க்கப்பட வேண்டுமா என்பதைத் தேர்வு செய்க.

   தரவுத்தள உரையாடல் பெட்டியில் தேர்வு செய்த விருப்பங்கள் சேமித்து வைக்கப்படுகின்றன. அடுத்த முறை உரையாடல் பெட்டியைப் பயன்படுத்தும் போது அவை செயல்பாட்டுக்கு அணியமாக உள்ளன. இந்த முறையில் 5 தரவுத்தளங்கள் வரை விருப்பங்களைச் சேமிக்கலாம்.

   தரவுகள் அட்டவணையாகச் செருகப்பட்டிருந்தால், தரவுகளுடன் அட்டவணை வடிவமைப்பு சேமிக்கப்படுவதில்லை. அட்டவணை அமைப்பைத் திரும்பப் பெற, **Auto Format** செயற்கூறினை தேர்வு செய்து அட்டவணையை அமைத்துக் கொள்ள வேண்டும். அப்படிச் செய்யும்போது **Star Office** அட்டவணை அமைப்பை முன்மாதிரியாக சேமித்துக் கொள்கிறது. அட்டவணையின் விருப்பங்கள் மாற்றப்படாத வரை இந்த முன்மாதிரி அட்டவணை அமைப்பு தானாக தரவுகளை அட்டவணையில் மீண்டும் சேர்க்க உதவுகிறது.

5. **>** அல்லது **>>** பொத்தான்களைப் பயன்படுத்தி குறிக்கப்பட்ட தரவுத்தள புலங்களை அட்டவணை நெடுவரிசைப் பட்டிப் பெட்டிக்குள் நகர்த்து.

>> இந்தப் பொத்தான் குறிக்கப்பட்ட அனைத்து தரவுத்தள புலங்களையும் அட்டவணை நெடுவரிசைப் பட்டிப் பெட்டிக்குள் நகர்த்த உதவுகிறது. பிறகு அட்டவணை நெடுவரிசைப் பட்டிப் பெட்டியில் குறிக்கப்பட்ட அனைத்து புலங்களும் ஆவணத்திற்குள் செருகப்படுகின்றன.

> இந்தப் பொத்தான் தரவுத்தளத்திலிருந்து தேர்வு செய்யப்பட்ட ஒரு புலத்தை அட்டவணை நெடுவரிசைப் பட்டி பெட்டிக்குள் நகர்த்துகிறது. புலத்தின் மீது இருமுறை கிளிக் செய்தும் அந்தப்புலத்தை அட்டவணை நெடுவரிசைப் பட்டி பெட்டிக்குள் நகர்த்தலாம். பிறகு அட்டவணை நெடுவரிசைப் பட்டிப் பெட்டியில் குறிக்கப்பட்ட அனைத்து புலங்களும் ஆவணத்திற்குள் செருகப்படுகின்றன.

6. அட்டவணைப் பகுதியில் உள்ள அம்புக்குறி பொத்தன்களைப் பயன்படுத்தி தரவுத்தள அட்டவணையிலிருந்து தேவையான நெடுவரிசைகளைத் தேர்வு செய்து உரை அட்டவணையில் சேர்க்கலாம்.

7. **Insert Table Heading** இணைப்பு, உரை அட்டவணையின் நெடுவரிசைகளுக்கு தலைப்பு கொடுக்க வழி செய்கிறது.

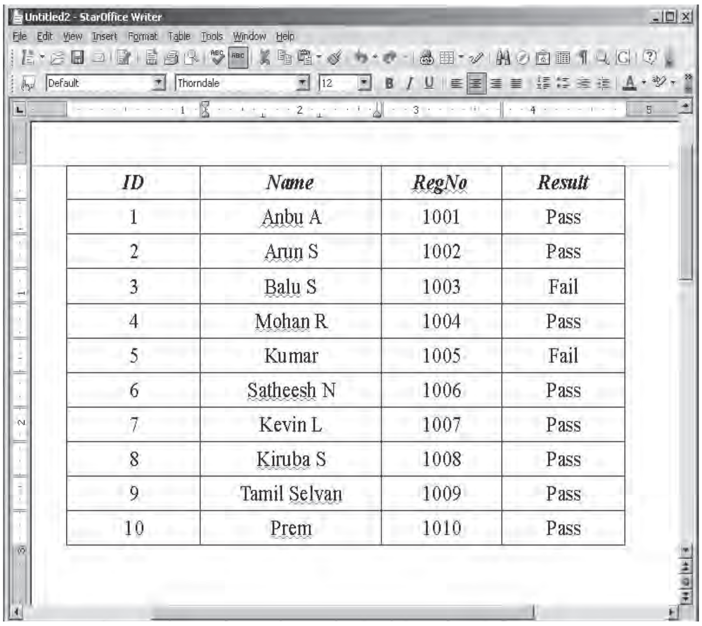
நெடுவரிசைகளுக்குப் பெயரிடு (Apply Column name)

உரை அட்டவணையின் நெடுவரிசைகளுக்குத் தலைப்புகளாக, தரவுத்தள அட்டவணையின் புலப் பெயர்கள் பயன்படுத்தப்படுகின்றன.

வரிசையை மட்டும் அமைக்க (Create row only)

இது உரை அட்டவணைக்குள் காலியாக உள்ள தலைப்பு வரியைச் செருகுகிறது. **Create row only** விருப்பத்தைப் பயன்படுத்தி, ஆவணங்களுக்குத் தலைப்புகளை இடலாம். இது தரவுத்தள புலங்களைச் சார்ந்ததாக இருக்கத் தேவையில்லை.

8. **OK** பொத்தானை கிளிக் செய்து உரை ஆவணத்தில் செருகப்பட்ட அட்டவணையைப் பார்க்கலாம். படம் 7.43 இதனைக் காட்டுகிறது.



ID	Name	RegNo	Result
1	Anbu A	1001	Pass
2	Arun S	1002	Pass
3	Balu S	1003	Fail
4	Mohan R	1004	Pass
5	Kumar	1005	Fail
6	Satheesh N	1006	Pass
7	Kevin L	1007	Pass
8	Kiruba S	1008	Pass
9	Tamil Selvan	1009	Pass
10	Prem	1010	Pass

படம் 7.43 Database table inserted in text document

சுருக்கம்:

தரவும் தகவலும்:

- தரவு என்பது மென்பொருள்களில் பயன்படும் மாறிகளின் மதிப்பை குறிக்கும். தரவுகள் தனித்துப் பொருள் தராத தகவல் கூறுகளாகும்.
- செயற்படுத்தப்பட்ட தரவுகளுக்கு இடையே நிலவும் உறவைத்தான் தகவல் என்கிறோம்.
- ஒரு தரவுத்தள மேலாண்மை அமைப்பு **(DBMS)** தரவுகளிலிருந்து தகவல்களை தொகுக்க உதவுகிறது

தரவுத்தளம்:

- ஒரு தரவுத்தளம் என்பது தொடர்புடைய தகவல்களின் அல்லது தகவல்களின் தொகுப்பைக் கொண்ட ஒரு களஞ்சியம் ஆகும்.
- ஒரு தரவுத்தள மேலாண்மை அமைப்பு என்பது பல பயனாளர்கள் தரவு தளத்தில் உள்ள தரவுகளை பெறவோ, தேவையெனில் மாற்றவோ அல்லது எளிய அல்லது சிக்கலான வினவல்களை செயல்படுத்துவதற்கோ தேவைப்படும் நிரல்களின் தொகுப்பாகும்.
- ஒரு தரவுத்தள அட்டவணையில் நெடுவரிசைகள் புலங்களையும், வரிசைகள் பதிவுகளையும் குறிக்கின்றன.
- புல வகைகள், உரை **(Text),** எண் **(Number),** ஏரணம் **(Logical),** போன்ற வகைகளைக் கொண்டுள்ளது.
- ஒற்றைக்கோப்பு **(FlatFile)** தரவுத்தளங்களில் பிற அட்டவணைகளுடன் தொடர்பு கொள்ள வழியில்லை. இவை சிறு வணிக பயன்பாடுகளுக்கு மட்டுமே பொருத்தமானதாகும்.
- உறவு நிலை **(Relational)** தரவுத்தளங்கள் அட்டவணைகளுக்கிடையே உறவுகளை ஏற்படுத்தும் திறமையை கொண்டதென்பதால் இவை பெரும் வலிமை பெற்றவையாக உள்ளன.
- படிநிலை **(Hierarchical)** மற்றும் வலை அமைப்பு **(Network)** தரவுத்தளங்கள் பழைய தத்துவங்களில் செயலாற்றும் தலைமைக் கணிப்பொறிகளில் பயன்படுத்தப்படுகின்றன.
- பொருள் நோக்கு **(object-Oriented)** தரவுத்தளங்கள், தரவு இயல்புகள் **(Characteristing)** தரவு பண்புகள், **(attributes)** தரவு செயல்பாடுகள் **(Procedure)** ஆகியவற்றை சிக்கலான பொருட்களாக **(object)** வரையறுக்கின்றன. இந்த பொருள்கள் **(objects)** மற்ற பொருள்களின் செய்திகள் மூலம் இணைந்து செயல்பட வல்லன.
- **Star Base ல் பணி புரிதல்**
- அட்டவணை அல்லது தரவுத்தாள் நோட்டம் **(Table and Data Sheet View)** அட்டவணையின் அமைப்பையும், பார்க்க விரும்பும் அட்டவணைகளின் பட்டியலையும் உருவாக்கவும் அட்டவணையில் தரவுகளை கொடுக்கவும் பதிப்பாய்வு செய்யவும். வேண்டிய அமைப்புகளை உருவாக்க பயன்படுத்தப்படுகின்றன.

- வடிகட்டிகள் சில நிபந்தனையின் கீழ் தேர்வுசெய்யப்பட்ட பதிவுகளை மேலோடி பார்க்க பயன்படுகின்றன.
- வரிசையாக்கம் **(Sorting)** ஒரு அட்டவணையில் உள்ள பதிவுகளை கொடுக்கப்பட்ட விதிமுறைப்படி ஒழுங்கு படுத்துகிறது.
- வினவல்கள் **(Queries)** பயனாளர் உருவாக்கிய கட்டளைகளாகும். இவை ஒன்று அல்லது அதற்கு மேற்பட்ட அட்டவணைகளிலிருந்து தரவுகளை, கொடுக்கப்பட்ட நிபந்தனைகளின் கீழ் தேர்வு செய்யவும் கையாளவும் பயன்படுகின்றன.
- **SQL** என்பது ஒரு சிறப்பு வினவல் மொழி ஆகும். இதனை வினவல் அமைப்பு மொழி **(Structured Query Language)** என்பர். இது மிக எளிய முறையில் தரவுத்தள அமைப்புகளுடன் தொடர்பு கொண்டு செயல்பட பயன்படுகின்றது.
- படிவங்கள் **(Forms)** பயனாளர்களால் வரையறுக்கப்பட்ட திரைகளாகும். இவை ஒரு அட்டவணையின் அல்லது வினவலின் தரவுகளை எளிய முறையில் கொடுக்கவும், பார்க்கவும், பதிப்பாய்வு செய்யவும் பயன்படுகின்றது.
- ஒரு அறிக்கை **(Report)** என்பது அச்சிடபட்ட தகவலாகும். இந்த அறிக்கையில் அச்சிடப்பட்ட தகவல் , வினவல்கள் மூலம், குறிப்பிட்ட நிபந்தனைகளை நிறைவு செய்து, தேவையான கணிப்பீடுகளையும் செய்து உருவாக்க பட்டவையாகும்.

பயிற்சி

I. கோடிட்ட இடத்தை நிரப்புக.

1. ஒரு அட்டவணையின் நெடுவரிசை ________________ ஐ குறிக்கின்றது.
2. பெயர்கள், முகவரிகள், தொலைபேசி எண்கள் போன்ற தரவுகள் ____________இல் தேக்கி வைக்கப்படுகின்றன.
3. ஒரு அட்டவணையின் வரிசை ________________ ஐ குறிக்கின்றது.
4. ஒரு தனி பதிவின் தரவுகளை காட்டும் திரைக்கு ______________________ என்று பெயர்.
5. பதிவுகளை அகர வரிசையிலோ அல்லது எண்களின் ஏற்ற இறக்க வரிசையிலோ ______________________ கொண்டு ஒழுங்கு செய்யலாம்.

II. கீழ்க்கண்ட வினாக்களுக்கு விடையளி

1. தரவுத்தளம் என்றால் என்ன ?
2. தரவுத்தள மேலாண்மை அமைப்பு என்றால் என்ன ?
3. தரவிற்கும், தகவலுக்கு உள்ள வேறுபாடு என்ன ?
4. தரவுத்தளத்தின் மூன்று உறுப்புகளைக் கூறு ?
5. ஒற்றைக் கோப்புத் தரவுத்தளம், உறவு நிலை தரவுத்தளத்திலிருந்து எவ்வாறு வேறுபடுகின்றது ?

6. எந்த தரவுத்தள அமைப்பில் பதிவுகள் பெற்றோர்– குழந்தை உறவு முறையைத் தழுவி இருக்கின்றது ?

7. பொருள் நோக்குத் தரவுத்தள அமைப்பில் பொருளை வரையறுக்கும் உறுப்புகளை பட்டியலிட்டு விவரி.

8. ஒரு தரவுத்தளத்தில் இருக்ககூடிய பல தரவு வகைகளை பட்டியலிடு.

9. தரவுத்தள பதிவுகளை வடிகட்டுதல் என்றால் என்ன ?

10. வினவல் என்றால் என்ன என்றும், அது எதற்காக பயன்படுகின்றது என்பதையும் விவரி.

அதிகாரம் 8

பல்லூடக அறிமுகம்

8.1 பல்லூடகம் என்றால் என்ன ?

பல்லூடகம் என்பது உரை, வரைபடம், அசைவுப்படம், ஒலி, ஒளிக்காட்சிகள் ஆகிய கூறுகள் அடங்கிய கணிப்பொறி தழுவிய ஒரு வழங்கு முறை. இந்தக் கூறுகளைக் கொண்டு செய்திகளை வழங்கும்போது அது பார்ப்பவரை, படிப்போரை ஈர்த்து நிறுத்துகிறது. பல்லூடக வழங்கு முறை பல வேறுபட்ட கல்வித் தேவைகளையும் கற்பிக்கும் பாணிகளையும் மேம்படுத்தி ஒரு புது தாக்கத்தை உண்டாக்குகிறது.

8.2 பல்லூடாகப் பயன்பாடுகள்

பல்லூடக வழங்கும் முறை பல்வேறுபட்ட துறைகளில் பயன்படுத்தப்படுகிறது. பொதுவாக பல்லூடக முறை பொழுதுபோக்கு, கல்வி, ஆய்வு மற்றும் வணிகத் துறைகளில் பெருமளவு பயன்படுத்தப்படுகிறது.

பொழுதுபோக்குத் துறையில், திரைப்படங்களுக்கு சிறப்பு மதிப்பூட்டும் காட்சிகளை அமைக்க பெருமளவில் பல்லூடக வசதிகள் பயன்படுத்தப்படுகின்றன. அசைவுப் படக் காட்சிகளையும் சித்திரக்கதைத் துணுக்குகள் ஆகியவற்றை உருவாக்கவும் பயன்படுத்தப்படுகின்றன. Finding Memo, Polar Express, Ice Age, Sindbad, Pandvas, Hanuman போன்ற அசைவுப்படங்கள் குழந்தைகள் மற்றும் பெரியோர்களிடையே மிகவும் புகழ் பெற்றவை. உயர்தர வரைபடங்கள் மற்றும் அசைவுப் படங்கள் போன்றவை வியப்பூட்டவல்ல இயல்பாகத் தோன்றும். கணிப்பொறி விளையாட்டுகள் உருவமைக்க அடிக்கடி பயன்படுத்தப்படுகின்றன. அப்படிப்பட்ட கணிப்பொறி விளையாட்டுகள் வீட்டுக் கணிப்பொறி அரங்குகளில் மிகப் பரவலாகக் காணப்படுகின்றன். விமானிகளுக்கு பயிற்சி அளிக்கப் பயன்படும் Flight Simulator இந்த வகையைச் சார்ந்த உயர் தொழில்நுட்ப படைப்பாகும்.

பல்லூடக செய்தி வழங்கும் அமைப்பு (Multimedia Messaging System), அல்லது MMS என்பது செல்பேசி மூலம் செய்திகளை அனுப்பவும் பெறவும் பயன்படும் மென்பொருள் ஆகும். இது நகைச்சுவைத் துணுக்குகள், இசை, அழைப்பொலிகள், படங்கள், சில நேரங்களில் ஒளிக்காட்சிகள் ஆகியவற்றை அனுப்பவும், பெறவும் பரவலாகப் பயன்படுத்தப்படுகிறது.

பல்லூடகம் கல்வி கற்பித்தலில் ஒரு பெரும் தாக்கத்தை உண்டாக்கி இருக்கிறது. வளர்ந்து வரும் பல்லூடக வசதிகளைப் பயன்படுத்தி பல உயர்தர மின்கல்வி மென்பொருள் தொகுப்புகள் உருவாக்கப் பட்டுள்ளன. அவை உலக அளவில் மாணவர்களை, பயிற்றுவிக்க பயன்படுத்தப் படுகின்றன. இவை இணையத்திலும் (WBT) குறுவட்டுகளிலும் (CBT) தானே கற்கும் வசதியுடன் வழங்கப்படுகின்றன. இணைய வழி அல்லது குறுவட்டுகளைப் பயன்படுத்தி கணிப்பொறியின் உதவியுடன் கற்கும் முறையை மின்வழிக் கல்வி என்றும் இணைய வழிக் கல்வி என்றும் கூறுவர். இம்முறையில் கற்க ஆசிரியர் தேவையில்லை. வகுப்பறை தேவையில்லை. மாணவர்களை ஒரு குறிப்பிட்ட நேரத்தில் குறிப்பிட்ட வகுப்பறைக்கு வர வேண்டிய தேவையில்லை. கற்க விரும்புவோர் எப்பொழுது வேண்டுமானாலும் எங்கிருந்து வேண்டுமானாலும் கற்கலாம். குறுவட்டு தருவிய (CBT) அல்லது இணையம் தழுவிய (WBT) கல்வி முறை, நிறுவங்களில் புதிதாக சேரும் பணியாளர்களுக்கு அறிமுகப் பயிற்சி, தொழிற்பயிற்சி, நிறுவனத்தில் பயன்படுத்தப்படும் மென்பொருள் பயன்பாட்டுப் பயிற்சி ஆகியவை வழங்க பெரிதும் பயன்படுகின்றன.

அச்சிட்ட புத்தக வடிவில் வழங்கும் முறையைவிட பல்லூடக வசதிகளைப் பயன்படுத்தி செய்திகளை வழங்கும் முறை பெரும் தாக்கத்தை உண்டாக்குகின்றது. எடுத்துக்காட்டாக, நீல் ஆம்ஸ்டாங் நிலவில் இறங்கும் காட்சியை ஒலிப் பின்னணியுடன் பார்க்கும் போது ஏற்படும் தாக்கம், அது பற்றி புத்தகத்தில் படிக்கும் போது ஏற்படுவது இல்லையே! e இணைப்பு (Hyper - Link) வசதியைப் பயன்படுத்தி வேண்டிய பகுதியை நாடிப் பார்க்க முடியும். மின்வழி/ இணையவழிக் கல்வி முறையில் வேண்டிய பொருள் பற்றி தேடிப் படிக்கும் வசதி உள்ளது. இந்த வசதி கல்வியை வேகமாகக் கற்க உதவுகிறது. மேலும் மிகுந்த பயன் அளிப்பதாகவும் உள்ளது.

பல்லூடகப் பயன்பாடு ஊடாடிப் படிக்க உதவுகிறது. எனவே இதனை ஊடாடும் பல்லூடகம் (Interactive Multimedia) என்பர். எடுத்துக்காட்டாக, சாசர் பல்லூடக விளையாட்டினை விளையாட்டுத்திடல் போகாமலேயே கணிப்பொறியிலேயே இயல்பாக விளையாடுவது போல விளையாடலாம். உருவாக்கம் ஒருவகை மாயை தான் ஆனால் அது உண்மையாகவே விளையாடுவது போன்ற ஒரு தாக்கத்தை உண்டாக்குகின்றது. இப்படிப்பட்ட சூழல்கள்,ஜாய்ஸ்டிக்ஸ் (Joy sticks) அல்லது உணலிகள் (Sensors) போன்ற உள்ளீட்டு கருவிகள் மற்றும் தலை அணி ஒலி வாங்கி, கண்ணாடி (Goggles) போன்ற வெளியீட்டுக் கருவிகள் ஆகியவற்றின் உதவியுடன் உண்டாக்கப்படுகின்றன. உண்மை நிகழ்ச்சிகளை மாயத்தோற்ற (Virtual Reality) முறையில் பார்க்க பல்லூடக வசதிகள் பெரிதும் பயன்படுகின்றன. இந்த முறை சில விளையாட்டுகளிலும், வான ஊர்தி ஓட்டுநர் பயிற்சிக்கும் பயன்படுகிறது.

பல்லூடக வசதிகள், நிறுவனங்கள் பற்றிய அறிமுக உரை வழங்குவதற்கு பெரிதும் பயன்படுகின்றன. பொருளுக்கேற்ற அசைவுப் படங்கள், பொருள்கள் மற்றும் கருவிகள் பற்றிய படங்கள், வரைபடங்கள், ஆகியவற்றின் துணையுடன் நிறுவனத்தை அறிமுகப்படுத்தும்போது அது பார்போரின் கவனத்தை ஈர்க்கின்றது. விளம்பரங்கள், பொருள்கள் பற்றிய அடைவுகள் (Product - catalogs), மின் இதழ்கள் ஆகியவற்றிலும் பல்லூடக வசதிகள் பயன்படுகின்றன.

பொறியியல், மருத்துவம், அறிவியல் ஆய்வு போன்ற பல துறைகளில் பல்லூடக வசதிகள் பயன்படுகின்றன. பொறியியல் துறையில் பொருள்களை வடிவமைக்கவும் சோதித்துப் பார்க்கவும் பயன்படுகிறது. மருத்துவத்தில், அறுவைச் சிகிச்சை போன்ற ஆபத்தான பணிகளில் மருத்துவர்களுக்கு மனிதர்களின் உடல்களைப் பயன்படுத்தாமலேயே மாய அறுவைச் சிகிச்சை (Virtual Surgery) மூலம் பயிற்சி அளிக்க பல்லூடகம் பயன்படுகிறது. அதே போன்று அறிவியல் அறிஞர்கள் அணு தொடர்பான மூலக்கூறுகள் பற்றிய ஆய்வில் ஈடுபடவும் பல்லூடகம் பெரிதும் பயன்படுகிறது.

8.3 பல்லூடகக் கூறுகள் : நிழற்படங்கள், ஒலி, அசைவுப்படங்கள் & ஒளிக்காட்சி

பல்லூடகம் ஒலி, நிழற்படங்கள், அசைவுப்படங்கள், ஒளிக்காட்சி, உரை போன்ற பல கூறுகளைக் கொண்டது. அவற்றைப் பற்றி இங்கு சுருக்கமாகப் பார்போம்.

நிழற்படங்கள் (Images)

நிழற்படங்களை கணிப்பொறியில் பல வடிவங்களில் வழங்கலாம். அவற்றில் சில GIF, JPG அசைவுப் படமாக்கப்பட்ட GIF, MPEG, NxView மற்றும் Shockwave கோப்புகளாகும்.

GIF மற்றும் JPG கோப்புகள் மிகுந்த அளவில் பயன்பாட்டில் உள்ளன. இவை நுண்ணுறைகளின் மதிப்புகளாகிய **0,1** என்ற குறியீடுகளால் சேமிக்கப்படும். கண்ணிப்பட (Bitmap) நிலை

நிழற்படங்களை (Static Images) குறுக்கி, குறுவடிவில் (Compresed Format) வழங்குகின்றன. புகைப்படங்கள் நிலை நிழற்பட வகையைச் சேர்ந்ததாகும். இரு கோப்புகளும் வெவ்வேறு குறுக்கு முறைகளைப் பயன்படுத்துகின்றன. GIF–இல் பயன்படுத்தப்படும் குறுக்கு முறை நிழற்படத்தை உருமாற்றமின்றி திரும்பத் தருகிறது. JPG முறை அப்படித் தருவதில்லை.

GIF (Graphic Interchange Format) முறை 8 கண்ணி வண்ணத்தட்டு (8-bit palette) அமைப்பிற்குள் அடங்கியது. எளிய குறைந்த வண்ணங்களைக் கொண்ட வரைபட நிழற்படங்களை கணிப்பொறியில் தேக்கி வைப்பதற்கு GIF முறை மிகவும் ஏற்றதாகும். எனினும் இது புகைப்படங்களைக் கையாளுவதற்கு ஏற்றதல்ல. இப்படிப்பட்ட பயன்பாடுகளுக்கு JPG, GIF முறையை விட சிறந்ததாகும். இருந்தாலும், ஏராளமான GIF நிழற்படங்கள் உள்ளதாலும், அசைவுப் படங்கள் உருவமைக்க GIF முறை வழிவகை செய்வதாலும், GIF முறை இன்னமும் இணையம் மற்றும் பிற பல்லூடக பயன்பாடுகளில் பெருமளவில் பயன்படுகிறது.

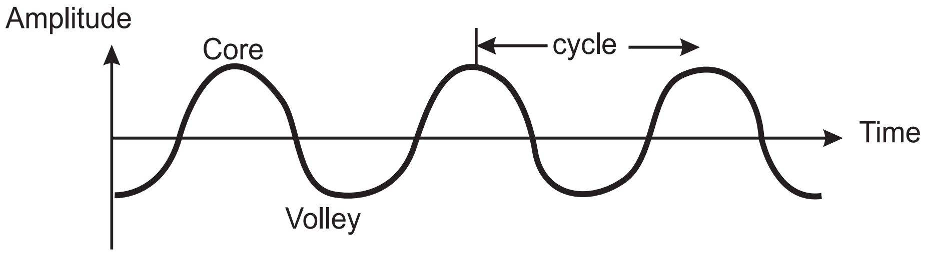
JPEG (Joint Photographic Experts Group) நிழற்படக் கோப்புகள் இழப்புடைய வடிவமைப்பைக் கொண்டதாகும். பெரும்பாலான புகைப்படக் கருவிகள் புகைப்படங்களை JPEG வடிவில் சேமித்து வைக்கும் வசதியைக் கொண்டுள்ளன. JPEG வண்ணத்தையும் வழங்கவல்லது. மேலும், பிற முறைகளை விட JPEG முறையில் கோப்பின் அளவு சிறியதாக இருக்கும்.

நிழற்பட கோப்புகளைக் குறுக்கி வழங்க இரு முறைகள் பயன்பாட்டில் உள்ளன. அவை இழப்புடைக் குறுக்கம் (Lossy Compression) இழப்பிலா குறுக்கம் (Lossless Compression) என்பனவாகும். இழப்புடை குறுக்க முறையில் ஏற்படும் இழப்பு கண்ணுக்கு புலப்படா அளவிற்கு மிகக் குறைந்த அளவே ஆகும். பெரும்பாலான நிழற்படங்களில் இழப்பு தெரிவதே இல்லை. JPEG இழப்புடைக் குறுக்கம் முறையில் செயல்படுவதாகும்.

நெறிய வரைபட முறை (Vectorgraphics) நிழற்படங்களைத் திறம்படத் திரையிலிட பயன்படுகிறது. மிகுந்த நிறுவனங்கள் நிழற்படங்களைத் திரையில் திறம்படக் காண்பிக்க நெறிய வரைபட அட்டைகள் (Vector graphics cards) வடிவமைத்து வழங்குகிறார்கள்.

ஒலி

நாம் கேட்கும் ஒலி தொடர் அலை உருவமைப்பைக் (Analog Wave Pattern) கொண்டவை. உச்ச மதிப்பு (Amplitude), அலை எண் (Frequency) ஆகிய இரண்டு பண்புகள் ஒலியின் தரத்தை வரையறுக்கின்றன. (படம்8.1). ஒலியின் அளவு அலையின் உச்ச மதிப்பைப் பொறுத்துள்ளது. ஒரு அலையின் நீட்டம், அதாவது அலையில் அடுத்தடுத்து உள்ள இரு உச்சப் புள்ளிகளுக்கிடையே உள்ள தொலைவு அலை அடர்த்தி (Wave) எனப்படுகிறது. ஒரு அலையின் நீட்டம் , அதாவது அலை அடர்த்தி மிகுதியாக ஆக ஆக ஒலியின் அளவு குறைகிறது. ஒலியை ஒரு பல்லூடகப் பயன்பாட்டில் சேர்க்க வேண்டுமெனில் தொடர் ஒலி அலைகளை எண்ணளவை வடிவில் (Digital Format) மாற்றி அமைக்க வேண்டும். இதனை இலக்க முறையாக்கல் (Digital Sampling) என்கிறோம். இவ்வாறு எண்ணளவை வடிவில் இலக்க முறைப்படுத்திய ஒலி அலையின் தரத்தை உயர்த்தலாம். எதிரொலி (echo) உள்மங்கல் (fade in), வெளி மங்கல் (fade out) போன்ற சிறப்புத் தாக்கங்களை, Sound Forge போன்ற ஒலி பாதிப்பு மென்பொருள்களைக் (Sound Edtiting Program) கொண்டு வடிவமைக்கலாம்.



படம் 8.1 Creating a Presentation from any Star Office Automation

ஒலியை பல்வேறுபட்ட வடிவங்களில் சேமித்துத் தேக்கி வைக்கலாம். இவை பற்றி இந்த அதிகாரத்தில் பின்னால் பயில்வோம்.

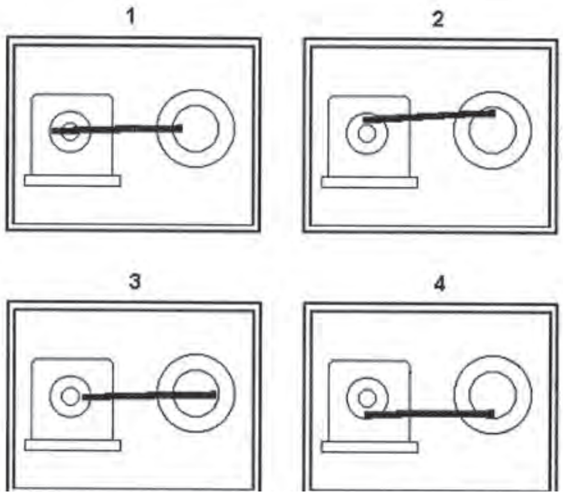
அசைவுப்படங்கள்

ஒரு தத்துவத்தை அல்லது கருத்தை தெளிவுபட விளங்குமாறு விளக்க அசைவுப்படங்கள் பெரிதும் பயன்படுத்தப் படுகின்றன. உண்மைக் காட்சிகள் அல்லது நிகழ்ச்சிகளை படமாக்கும் ஒளிக்காட்சிகள் போலன்றி அசைவுப்படக் காட்சி வரைபடங்கள், ஓவியங்கள் ஆகியவற்றைக் கொண்டு படமாக்கப்படுகிறது.

அசைவுப்படங்கள் இரு பரிமாணம் (Two - dimensional) அல்லது முப்பரிமாணங்களைக் (Three - dimensional) கொண்டதாக இருக்கலாம். இரு பரிமாண அசைவுப்படங்களை உருவமைக்கும் முறைகளின் அடிப்படையில் அவற்றைக் கீழ்க்காணும் இரு வகைகளாகப் பகுக்கலாம்.

- கண்ணிகள் தழுவிய (Cell - based) அசைவுப் படங்கள்
- பொருள் தருவிய (object - based) அசைவுப் படங்கள்

கண்ணிகள் தழுவிய அசைவுப்படம் பல வரைபடங்களைக் கொண்டு உருவாக்கப்படுகிறது. (படம் 8.2). ஒவ்வொரு வரை படமும் ஒன்றுக்கொன்று சற்று வேறுபட்டு இருக்கும். அவை ஒரு குறிப்பிட்ட வேகத்தில் நகர்த்தப்படும் போது அசைவுப் பொருள் இயல்பாக செயல்படுவது போலத் தோன்றுகிறது.

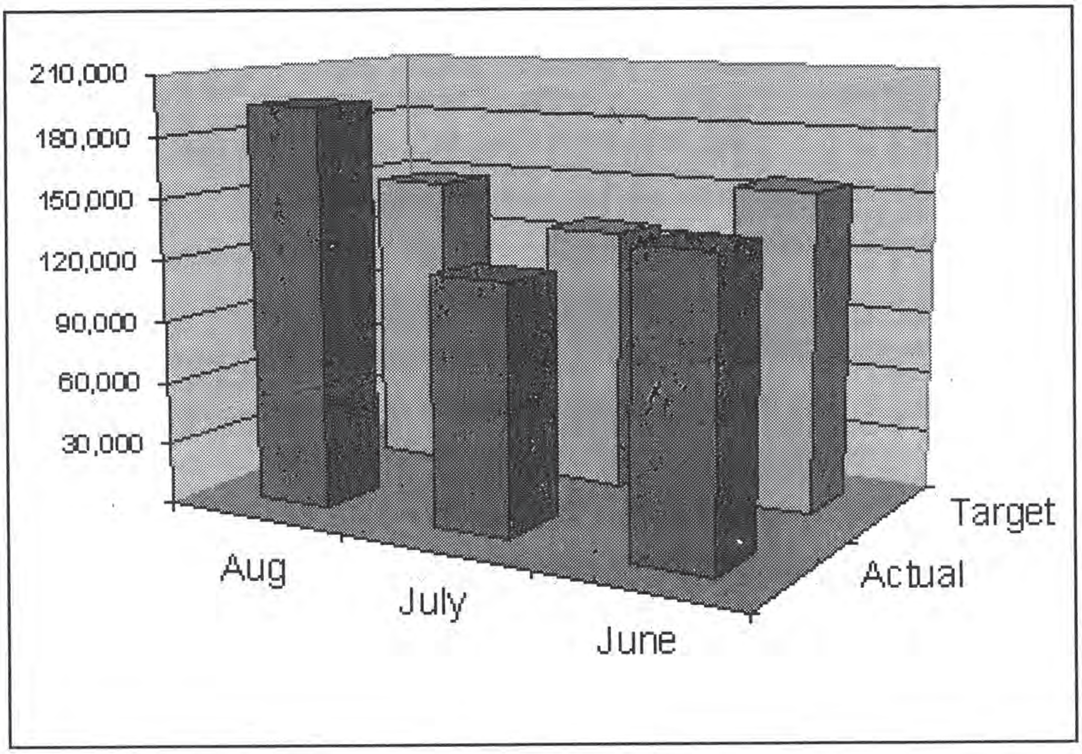


படம் 8.2 Cell - based Animation

பொருள் தழுவிய அசைவுப் படங்கள் சில்லு (Slide) அல்லது பாதை (Path) அசைவுப் படங்கள் என்றும் அழைக்கப்படுகின்றன. இவை பொருள்களை திரைக்கு முன் நகர்த்தி உருவாக்கப்படுகின்றன. இப்படிப்பட்ட அசைவுப் படங்கள் கணிப்பொறி விளையாட்டுகளில் பயன்படுத்தப்படுகின்றன. திரையில் பந்து ஓடுவதைக் காட்டுவதை ஒரு எடுத்துக்காட்டாகக் கொள்ளலாம்.

முப்பரிமாண அசைவுப் படங்கள் மூன்று நிலைகளில் உருவமைக்கப்படுகின்றன. அவையாவன,

1. படிமமாக்கல் (Modeling)
2. அசைவாக்கல் (Animating)
3. வழங்குதல் (Rendering)



படம் 8.3 **A 3-D Image**

முப்பரிமாண அசைவுப்படங்களை உருவாக்க முதலில் பொருளின் முப்பரிமாண வடிவமைப்பை உருவாக்க வேண்டும். பிறகு அசைவூட்டலுக்குத் தேவையான படிப்படியாக மாற்றங்களை கொண்ட பல வரைபடங்களை வரைந்து கொள்ள வேண்டும். இறுதியாக பொருளின் பண்புகளை வெளிப்படுத்தும் வண்ணம், இழைவு (Texure) போன்றவற்றை வழங்க வேண்டும். அசைவுப்படத்தை உருவாக்கிய பிறகு, உருமாற்றம் செய்தல் (Morphing) உருக்குலைத்தல் (Warping) போன்ற செயற்பாடுகள் மூலம், அதனை சிறப்புத் தாக்கங்களுடன் மேலும் மேம்படுத்தலாம். உருமாற்றத் தொழில்நுட்பம் என்பது இரண்டு அல்லது அதற்கு மேற்பட்ட நிழற்படங்களை ஒன்று சேர்த்து ஒரு புது நிழற்படத்தை வடிவமைக்கும் முறையாகும். உருக்குலைத்தல் (Warping) என்பது ஒரு நிழற்படத்தை திருத்தி –உருமாற்றி வேறு ஒரு நிழற்படமாக மாற்றும் முறையாகும். பெரும்பாலான இக்கால பல்லூடகப் பயன்பாடுகள், குறிப்பாக விளையாட்டுகள், இந்தத் தொழில்நுட்பங்களை மாயத் தோற்ற (Vitual reality)

தொழில்நுட்பத்தோடு இணைத்து, பார்ப்போர் தானும் அந்தச் சூழலில் அங்கமாக வேண்டும் வகையில் உருவமைக்கப்படுகின்றன.

ஒளிக்காட்சி (Video)

ஒலியைப் போல ஒளிக்காட்சியும் பதிவு செய்யப்பட்டு தொடர் நிகழ்ச்சியாகத் திருப்பிக் காண்பிக்கப்படுகிறது. எனவே, பல்லூடகப் பயன்பாட்டிற்கு ஒளிக்காட்சிக் குறிப்புகளும் எண் குறிப்பமைப்பாக (digital formats) மாற்றப்பட வேண்டும்.

ஒளிக்காட்சி கோப்புகள் மிகவும் பெரிதாக இருக்கும், கோப்பின் அளவு, படச்சட்டங்கள் ஓடும் வேகம் (frame rate) நிழற்படத்தின் அளவு, வண்ணங்களின் அளவு ஆகியற்றைப் பொறுத்துள்ளது. இவற்றை கட்டுப்படுத்தி ஒளிக்காட்சிக் கோப்பின் அளவை ஏற்புடைய அளவிற்கு அமைத்துக் கொள்ளலாம். எனினும், 256 வண்ணங்களுக்குக் குறைவாக உள்ள ஒளிக்காட்சிகள் இருளடைந்து காணப்படும். இதே போல படச் சட்டங்கள் ஓடும் வேகம் விநாடிக்கு 15 சட்டங்களுக்குக் குறைவாக இருக்குமானால் ஒளிக்காட்சி ஆடத் தொடங்கிவிடும்.

ஒளிக்காட்சி கோப்புகள் அளவை குறுக்கல் (Compression) முறையைப் பயன்படுத்தியும் குறைக்கலாம். குறுக்கல் என்பது, கோப்பில் பலமுறை வரும் உருக்களை கண்டறிந்து அவற்றில் ஒன்றை மட்டுமே தேக்கி வைக்கும் தொழில்நுட்பமாகும். இருவகையான குறுக்கல் முறைகள் உள்ளன.

- இழப்பிலா குறுக்கம்
- இழப்புடை குறுக்கம்

இழப்பிலா குறுக்க முறையில், குறுக்கித் திரும்பப் பெறும் ஒளிக்காட்சி மூலக்காட்சி போலவே இழப்பின்றி தோன்றும் இழப்புடைய குறுக்க முறையில், குறுக்கித் திரும்பப் பெறும் ஒளிக்காட்சிகள் தரம் ஓரளவு குறைந்து காணப்படும்.

வழக்கு மிகு பல்லூடகப் படிவங்கள்

மிகுந்த வழக்கில் உள்ள பல்லூடகப் படிவங்கள் பற்றி இங்கே பார்ப்போம்.

MIDI படிவம்

எண் இடைமுக இசைக்கருவிப் (Musical Instrument Digital Intreface - MIDI) படிவம், கணிப்பொறியில் உள்ள ஒலி அட்டை (Sound Card) அல்லது தொகுப்பிக்கு (Synthesizer) மின்னிசைக் கருவிகளின் இசையை கொண்டு சேர்க்க இந்தப்படிவம் பயன்படுகிறது.

MIDI படிவமுறை 1982–இல் கண்டுபிடிக்கப்பட்டது. பலவகைப்பட்ட இசைகளை பல்லூடகத்தில் பயன்ப்டுத்த இம்முறை பரவலாகக் கையாளப்படுகிறது. MIDI படிவம் இசையை இசை வடிவில் கொள்வதில்லை. எண் வடிவில் மட்டுமே கொள்ளக்கூடியது. எனவே இக்கோப்புகளின் அளவு குறைந்த அளவில் இருக்கும். எடுத்துக்காட்டாக 5 நிமிட இசைக் கோப்பின் அளவு 25 கிலோபைட்க்கு உட்பட்டதாகவே இருக்கும். புகழ் பெற்ற உலவிகள் (Browsers) முதற்கொண்டு பெரும்பாலான மென்பொருள்கள் இந்தப் படிவத்தை பயன்படுத்துகின்றன.

MIDI படிவங்களின் கோப்புகள் .midi அல்லது .mid என்னும் விரிவுகளைக் கொண்டுள்ளன.

Real Audio /Real Video படிவம்

Real Audio / Real Videoபடிவம் **1995–**இல் Real Networks என்னும் நிறுவனத்தால் உருவாக்கப்பட்டது. இது ஒலி மற்றும் ஒளிக்காட்சிகளை வழங்க உதவுகிறது. இந்த முறை இணையத்தில் ஒலி / ஒளிக்காட்சி கோப்புகளை வழங்க பெருமளவில் பயன்படுத்தப்படுகிறது. மிகவும் குறைந்த தட அகலம் (Bandwidth) கொண்ட இணையத் தொடர்பிலும் ஒலி/ ஒளிக்காட்சிகளை அனுப்ப இம்முறை பயன்படுகிறது. எனினும் இம்முறையின் தரம் சற்று குறைந்தே காணப்படுகிறது. இக்கோப்புகள் .rm அல்லது .ram என்னும் விரிவுகளைக் கொண்டுள்ளது.

AU படிவம்

இது இன்னொரு வகை ஒலிப்படிவம். இது புகழ்பெற்ற பல மென்பொருள்களில் பயன்படுத்தப்படுகின்றன. இதன் கோப்புகள் .au என்னும் விரிவைக் கொண்டுள்ளது.

AIFF படிவம்

ஒலிப் பரிமாற்றக் கோப்புப் படிவம் (Audio Interchange File Format - AIFF) Apple நிறுவனத்தால் உருவமைக்கப்பட்டது. இது ஒரு இயக்க மென்பொருள் தளத்திலிருந்து இன்னொரு இயக்க மென்பொருள் தளத்திற்கு மாறி செயல்படக் கூடியது. அல்லது எல்லா உலவிகளுக்கும் ஏற்றது அல்ல. எனவே அவ்வளவாக வழக்கில் இல்லை. இதன் கோப்புகள் .aif அல்லது .aiff என்னும் விரிவுகளைக் கொண்டுள்ளது.

SND படிவம்

ஒலி அல்லது SND படிவமும் Apple நிறுவனத்தால் உருவமைக்கப்பட்டது AIFF படிவம் போன்றே ஒன்றிலிருந்து மற்றொரு இயக்க மென்பொருள் தளத்திற்கு மாறி செயல்படும் ஆற்றல் அற்றது. பல புகழ் பெற்ற உலவிகளுக்கும் ஏற்றதல்ல. எனவே இது அவ்வளவாக வழக்கில் இல்லை. இதன் கோப்புகள் .snd என்னும் விரிவைக் கொண்டுள்ளது.

Wave படிவம்

Wave படிவம் IBM மற்றும் Microsoft நிறுவனங்களால் உருவமைக்கப்பட்டது. இது எல்லா வகையான Windows இயக்க மென்பொருள் தளங்களிலும், எல்லா புகழ்பெற்ற உலவிகளிலும் செயல்படக் கூடியவை. எனவே இது மிகவும் புகழ்பெற்ற படிவமாகும். இதன் கோப்புகள் .Wav என்னும் விரிவைக் கொண்டுள்ளது.

MP3 / MPEG படிவம்

MPEG படிவம் Moving Pictures Experts Group என்னும் நிறுவனத்தால் உருவமைக்கப்பட்டது. MPEG படிவம் ஒளிக்காட்சி கோப்புகளுக்கும் MP3 ஒலிக் கோப்புகளுக்கும் உருவமைக்கப்பட்டன. இவை இணையத்தில் மிகப் பரவலாகப் பயன்பட்டு வருகின்றன. இதற்குக் காரணம் இவை உயர் தரத்துடன் கூடிய நல்ல குறுக்குத்திறன் கொண்டவை. மேலும் இவை எல்லா இயக்க மென்பொருள்களிலும் எல்லா உலவிகளிலும் செயல்படக் கூடியவை. MP3 படிவத்தில் ஒலிக் கோப்புகள் .mp3 அல்லது .mpga என்னும் விரிவுகளுடனும் MPEG படிவத்தில் ஒளிக்காட்சி கோப்புகள் .mpg அல்லது .mpeg என்னும் விரிவுகளிலும் தேக்கி வைக்கப்படுகின்றன.

AVI படிவம்

ஒலி– ஒளி இடைப் பின்னிய (Audio Video Inerleave) படிவம் 1992–இல் Microsoft நிறுவனத்தால் உருவமைக்கப்பட்டது. இது பெரும்பாலான உலவிகளிலும் எல்லா வகையான Windows இயக்க மென்பொருள் தளங்களிலும் செயல்படக் கூடியவை. இது இணையத்தில் மிகப் பரவலாக பயன்படுகிறது. இதில் ஒளிக்காட்சி கோப்புகள் .avi என்ற விரிவில் தேக்கப்படுகின்றன.

Windows Media படிவம்

Windows Media படிவம் Microsoft நிறுவனத்தால் உருவமைக்கப்பட்டது. இதுவும் இணையத்திலும் Windows இயக்க மென்பொருள் தளங்களிலும் பரவலாக பயன்படுத்தப்படுகிறது. இதன் கோப்புகள் .wmv என்னும் விரிவைக் கொண்டுள்ளது.

Qucik Time படிவம்

Qucik Time படிவம் Apple நிறுவனத்தால் ஒளிக்காட்சி கோப்புகளைத் தேக்கி வைக்கும் முதன்மை நோக்கத்துடன் உருவமைக்கப்பட்டது. இதன் கோப்புகள் .mov என்னும் விரிவைக் கொண்டுள்ளது.

Shock wave படிவம்

Shock wave படிவம் Macromedia என்னும் நிறுவனத்தால், Flash மென்பொருளைப் பயன்படுத்தி வடிவமைக்கும் பல்லூடகக் கோப்புகளைத் தேக்கி வைக்க உருவமைக்கப்பட்டது. இந்தக் கோப்புகளை செயல்படுத்த துணைக் கருவிகள் தேவை. எனினும் இந்தத் துணைக் கருவிகள் Netscape மற்றும் Internet Explorer உலவிகளில் முன்னிடப்பட்டு வருகின்றன. இதன் கோப்புகள் .swf என்னும் விரிவைக் கொண்டுள்ளது.

பல்லூடக வன்பொருளும் மென்பொருளும்

பெரும்பாலான கணிப்பொறிகள் பல்லூடகக் கோப்புகளைக் கையாளப் போதுமானவை. எனினும் கூடுதலான நினைவகம், மிகுந்த திறன் கொண்ட ஒலி/ஒளி அட்டைகள், மிகுந்த பிரிதிறன் (Resolution) கொண்ட திரையகங்கள் (Monitors) போன்றவற்றால் பல்லூடக நிகழ்ச்சிகளை மேலும் சிறப்பாகக் காண இயலும். மேலும் Windows Media Player போன்ற சிறப்பு மென்பொருள்களின் உதவியுடன் பல்லூடக நிகழ்ச்சிகளை மேலும் சிறப்பாகக் காண முடியும்.

பல்லூடகக் கோப்புகளை உருவமைக்க சிறப்பு மென்பொருள்கள் தேவை. வழக்கில் கிடைக்கும் மென்பொருள்களில் சில Flash, Dreamweaver, Maya ஆகியவை மிகுந்த வசதிகளை கொண்ட மென்பொருள்களும் உள்ளன.

உள் ஒலி – ஒளிக்காட்சி

பல்லூடகத்தில் உள்ள ஒலி/ ஒளிக் காட்சிகளை இணையதளத்திற்குள் இருந்தோ அல்லது வெளிக்கருவி துணையுடனோ ஒட்டலாம். HTML ஒட்டிகளைப் (Tag) பொறுத்து ஒட்டும் முறை அமைகின்றது. ஒலி/ ஒளிக் காட்சி கோப்புகளை இணையப் பக்கத்தின் ஒரு பகுதியாக வழங்கினால் இவை வெளிக்கருவியின் துணையின்றி தானே ஓடும். இதனை உள் ஒலி (Inline Audio) உள் ஒளி (Inline Video) என்பர். ஒரு இணைய பக்கத்தில் உள் ஒலி வசதியை

<bgsound> மற்றும் <img> ஒட்டிகளைப் (Tags) பயன்படுத்தி உருவாக்கலாம். ஒலி/ ஒளிக் காட்சிகளை ஓட்டுவதற்கு உலவியின் உதவியால் நிறுவப்படும் மென்பொருளை Plug-in என்று அழைக்கிறோம். Plug -in போன்ற துணைப் பயன்பாடுகள் ஒலி/ ஒளிக் காட்சிக் கோப்புகளின் அமைவுகளைக் (Settings) கட்டுப்படுத்த உதவுகின்றன. எடுத்துக்காட்டாக, ஒலி அளவைக் கட்டுப்படுத்துதல், பின்னிடல் (new__) ஓட்டுதல் (play), இடைநிறுத்தல் (Pause), நிறுத்தல் (Stop) போன்ற கட்டுப்பாடுகளை பார்ப்பவர்களே செய்து கொள்ளலாம்.

துணைப் பயன்பாடுகளை <embed>, <applet> அல்லது <object> போன்ற ஒட்டிகளின் உதவியால் நிறுவிக் கொள்ளலாம்.

குறிப்பு : இந்த ஒட்டிகள் HTML / XHTML தர ஒட்டிகள் அல்ல. சில ஒட்டிகள் Netscape உலவியாலும் சில Internet Explorer உலவியாலும் ஏற்கப்பட்டுள்ளன. நாமே ஒரு ஊடகக் கோப்பில் e–இணைப்புகளை (Hyperlinks) அமைக்கலாம். அப்படிச் செய்யும் போது உலவி, Window Media Player போன்ற துணைப் பயன்பாட்டினை தானே இட்டு கோப்புகளை ஓட்ட உதவும்.

8.4 பல்லூடகப் பயன்பாடு

பல்லூடகக் கூறுகளைச் சேர்த்து செய்திகளை வழங்கும் போது அது மிகுந்த தாக்கத்தை உண்டாக்குகின்றது. எனினும் பல்லூடகக் கூறுகளை சேர்க்கும்போது கோப்பின் அளவு மிகுந்து அளவு கூடும் என்பதை அறிவீர்கள். அத்துடன் ஊடகத்தின் குறைபாடுகளும் சேர்ந்து வழங்கு பொருளை பார்க்கவே இயலாமல் போகலாம். அதனால் பார்ப்போர் ஆர்வம் குறையும். இந்நிலை ஏற்படாமல் இருக்க பல்லூடக வழங்கு பொருளை உருவமைக்கும்போது கீழ்க்காணும் குறிப்புகளைக் கவனத்தில் கொள்க:

- பல்லூடக வசதியுடன் வழங்க வேண்டிய தன் காரணத்தை நன்கு ஆய்ந்து கட்டாயம் தேவை தானா என்று சிந்தித்து பயனற்றதைத் தவிர்க்க வேண்டும்.
- கோப்பின் அளவு பெரிதாக இருக்குமானால் மாற்று பல்லூடக வசதிகளைப் பயன்படுத்துவது பற்றி சிந்திக்கலாம். எடுத்துக்காட்டாக, ஒளிக்காட்சியாக வழங்கினால் கோப்பின் அளவு பெரிதாகுமெனில், அதனை அசைவுப் படமாக்கவோ அல்லது வெறும் படத்துடன் ஒலியைச் சேர்த்தோ வழங்கலாம்.
- பெரிய அளவைக் கொண்ட ஒலி/ ஒளிக்காட்சி கோப்புகளை தொடரோடச் (Sreaming) செய்யலாம். HTTP செயல்பாட்டில் கோப்புகளை பொறிக்குள் முழுமையாக இறக்கப்பட்ட பின்னரே கோப்பு ஓடத் தொடங்குகிறது. தொடரோடச் செய்வதில் பயனாளர் கணிப்பொறி இணையத்தில் உள்ள சேவைக் கணிப்புடன் கோப்பு ஓடி முடிகின்ற வரை தொடர்ந்து தொடர்பு கொண்டிருக்கும்.
- பல்லூடகக் கோப்பின் கூறுகளை பார்வையாளர் கட்டுப்படுத்துகின்ற வகையில் ‘பார்வையர் கட்டுப்பாடுகள்’ வழங்க வேண்டும். எடுத்துக்காட்டாக தவிர்ப் பொத்தான் (Skip button) ஏற்படுத்தினால் பல்லூடகக் கோப்பினை ஓட்டிப் பார்க்கும் போது வேண்டாத பகுதிகளைத் தவிக்க வசதியாக இருக்கும்.
- கோப்புகளை கணிப்பொறியில் இறங்கும் போது எந்த அளவு கோப்பு இறங்கியுள்ளது என்பதையும் மீதியுள்ள கோப்பு இறங்க இன்னும் எவ்வளவு நேரம் ஆகும் என்பதையும் அறிந்து கொள்ளும் வகையில் பார்வையர் பின்னூட்டு (View feed back) பொத்தான் கொடுத்தால், பார்வையர் தங்கள் நேரத்தை நன்கு பயன்படுத்திக் கொள்ள இயலும்.

- பல்லூடகக் கோப்புகளின் கூறுகளை, குறிப்பாக ஒளிக்காட்சிக் கூறுகளை, சிறிய சன்னல் திரைகளில் வழங்க வேண்டும்.

சுருக்கம்:

- பல்லூடகப் பயன்பாடு நாள்தோறும் பெருகிக் கொண்டு வருகின்றது.
- வழங்கு பொருளின் தரத்தை உயர்த்த பல்லூடகப் பயன்பாடுகள் உரை, ஒலி, ஒளிக்காட்சி போன்ற கூறுகளைக் கொண்டுள்ளன.
- பல்லூடக வசதிகள் கல்வி, பொழுதுபோக்கு, நிகழ்த்தல் போன்ற செயல்பாடுகளில் பேரளவில் பயன்படுகின்றன.
- பல்லூடகப் பயன்பாடுகளைச் செயல்படுத்த தனிச்சிறப்புடைய வன்பொருள்களும் கிடைக்கின்றன.
- ஒலி, ஒளிக்காட்சி ஆகியவற்றை பல்லூடகக் கூறுகளாக வழங்க பல வகையான படிவங்கள் உள்ளன.
- இந்தப் படிவங்கள் இணையப் பக்கங்களைக் கவர்ச்சி உள்ளவையாக உருவமைக்கப் பயன்படுகின்றன.
- பல்லூடக பொருட்கூறுகளை உருவமைத்தல் மிகவும் போராட்டத்திற்குரிய செயலாகும்.

பயிற்சி

1. உரை, ஒலி, நிழற்படங்கள் மற்றும் ஒளிக்காட்சிகள் கொண்ட HTML பக்கத்தை வடிவமைக்கவும்.

   இத்துடன் கொடுத்துள்ள குறுவட்டில் (CD) உள்ள ஒலிக்கோப்புகள் நிழற்படக் கோப்புகள், ஒளிக்காட்சிக் கோப்புகள் ஆகியவற்றைக் கவனிக்கவும்.

அதிகாரம் 9

நிகழ்த்துதல் (Presentation)

9.1 அறிமுகம்

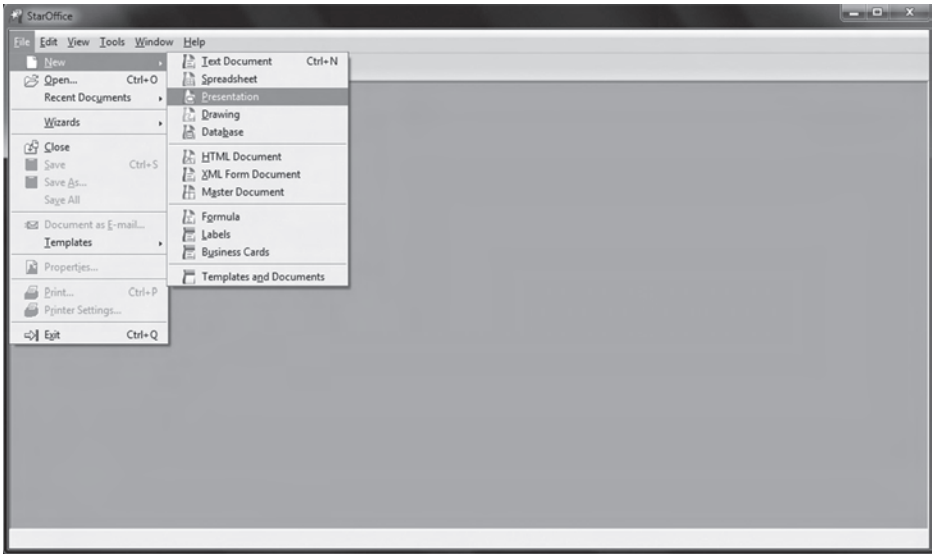
அவையோர்க்கு அல்லது கேட்போர்க்கு செய்தியை வழங்கப் பயன்படுத்தப்படும் பல்வேறு முறைகளில் கணிப்பொறி மூலம் நிகழ்த்தும் செய்திச் சில்லுகள் (Slide Presentation) முறை மிகுந்த தாக்கத்தை உண்டாக்குகின்றது. சில்லு நிகழ்த்த (Slide Presentation) முறையில் உரை, ஒலி, கையால் வரைந்த நிழற்படங்கள், ஒளிக்காட்சிகள் போன்ற பல்வேறு ஊடகச் செய்திகளுடன் வழங்கலாம். வரைபடங்கள், அட்டவணைகள், பின்னணிப்படங்கள், அசைவுப்பட சில்லுகள் ஆகியவற்றால் நிகழ்த்துதலை மேலும் மேம்படுத்தலாம்.

Star Office – இன் StarOffice impress நிகழ்த்துகலை வழங்குவதற்கான வசதிகளைக் கொண்டுள்ளது Star Office impress நிகழ்த்தல் சில்லுகளை (Presentation Slides) சேமித்துத் தேக்கி வைக்க, மாற்றி அமைக்க, அச்சிட, சில்லுகளை காட்சிகளாகப் பார்க்க HTML பக்கங்களாக்க போன்ற பல்வேறு செயல்களைச் செய்வதற்கான கருவிகளைக் கொண்டுள்ளது. இது Microsoft வழங்கும் Power Point Presentation சில்லுகளை Star Office impress –க்குள் இறக்கி மாற்றி அமைக்க வழி செய்கிறது.

9.2 ஒரு அடிப்படை நிகழ்த்தல்

கீழ்க்காணும் பல வழிகளில் StarOffice impress மென்பொருளைப் பயன்படுத்தி நிகழ்த்தல் உருவமைக்கலாம்.

- Star Office பயன்பாட்டிற்குள் இருக்கும்போது File→New→Presentation என்னும் பொத்தான்களை அழுத்தி உருவமைக்கலாம். (படம் 9.1)



படம் 9.1 Creating a Presentation from any StarOffice Application

- Star Office –க்கு வெளியில் இருக்கும்போது,Start→ All Programs→ Star Office 8 →StarOffice impress என்ற பொத்தான்களை கிளிக் செய்து StarOffice- → நிறுவி நிகழ்த்தல் உருவமைக்கலாம்.

9.2.1 நிகழ்த்தல் கருவியால் நிகழ்த்தல் அமைத்தல்:

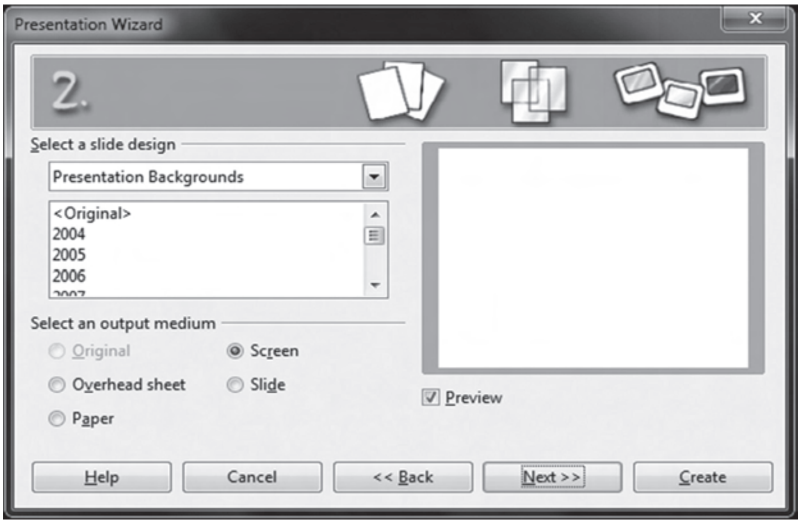
புதிய நிகழ்த்தல் (Presentation) ஒன்றை அமைக்கும்பொழுது, நிகழ்த்தல் கருவி (Presentation wizard) படம் 9.2–இல் உள்ளது போல் தானே தோன்றும்.

படம் 9.2 Choosing the Type of the Presentation

இந்தக்கருவி நிகழ்த்தல் சில்லுகளை வடிவமைக்க வழிகாட்டும். முதல் பக்கத்தில் மூன்று மாற்று முறைகள் கொடுக்கப்பட்டுள்ளன :

1. காலி நிகழ்த்தல் சில்லு ஒன்றை ஏற்படுத்தல்.
2. முன்பே உருவமைக்கப்பட்ட நிகழ்த்தல் சில்லுகளைக் கொண்ட கோப்பிலிருந்து தேர்வு செய்தல்.
3. ஏற்கனவே உருவமைத்து சேமித்து வைத்துள்ள நிகழ்த்தல் சில்லுகளைத் திறத்தல்.

காலி நிகழ்த்தல் சில்லுவைத் தேர்வு செய்து நிகழ்த்தல் சில்லு உருவமைக்க, Empty Presentation –ஐத் தேர்வு செய்து Next பொத்தானை கிளிக் செய்தால் படம் 9.3 –இல் காட்டியுள்ள பக்கம் தோன்றும்.

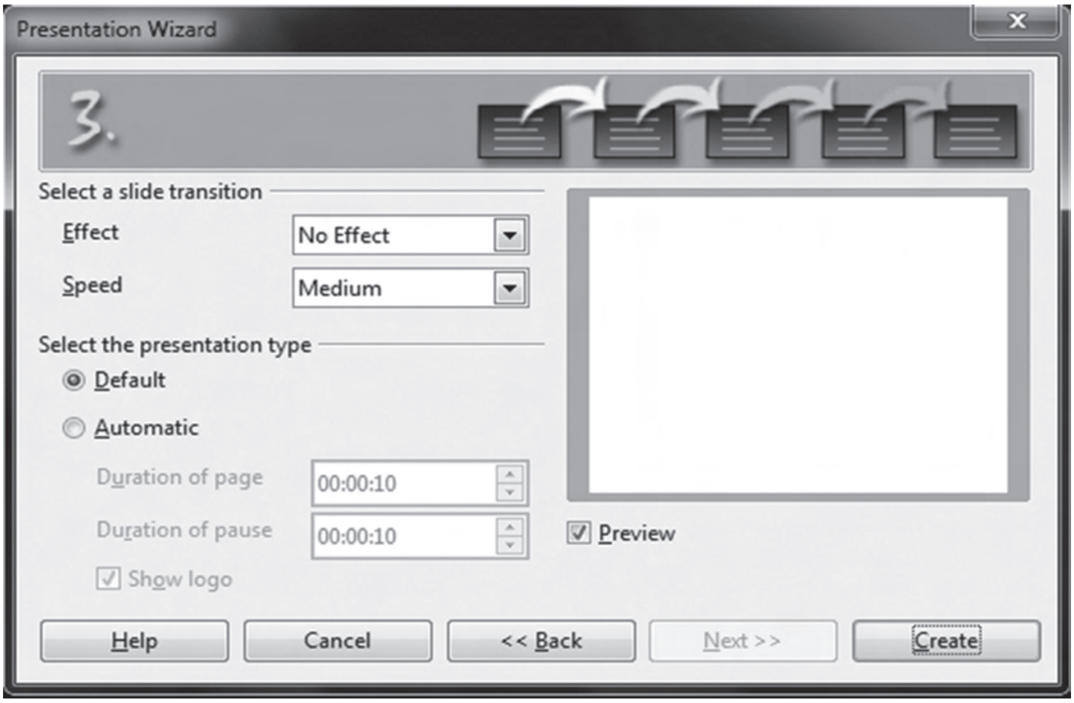


படம் 9.3 Choosing the Background and Output Medium of the Presentation

இந்தப் பக்கத்திலிருந்து, நிகழ்த்தல் சில்லுகள் எல்லாவற்றுக்கும் தேவையான பின்னணித் திரையைத் தேர்வு செய்து கொள்ளலாம். இப்பக்கத்தின் இடப்புறம் உள்ள ஒரு இழுபட்டியல் (Drop - down list) மூன்று வகை பின்னணித் திரைகளை வழங்குகின்றன. பின்னணி வகைப்பட்டிருக்கும் கீழ் ஒரு பட்டியல் பெட்டி அந்த வகையில் பல்வேறு பின்னணிகளைக் காட்டுகிறது. இந்தப் பின்னணிச் சில்லுகளில் தேர்வு செய்த சில்லுவை முன் கூட்டியே பார்க்கும் வகையில் இப்பக்கத்தின் வலப்புறத்தில் முன்காட்சிப் பெட்டி ஒன்று உள்ளது. Preview பொத்தான் கிளிக் செய்து இவ்வசதியைப் பெறலாம்.

Select an output medium என்ற இணைப்பு இப்பக்கத்தின் கீழ்ப்பகுதியில் உள்ளது. இதன் மூலம் தேவையான ஊடகத்தைத் தேர்வு செய்து கொள்ளலாம்.

தேவையான ஒரு பின்னணியைத் தேர்வு செய்தபின் Next பொத்தானை கிளிக் செய்யவும். தேர்வு செய்த பின்னணியை தேவையானால் எப்பொழுது வேண்டுமானாலும் மாற்றிக் கொள்ளலாம். Next பொத்தானை கிளிக் செய்ததும் படம் 9.4 காட்டியுள்ள மூன்றாவது பக்கம் தோன்றும்.



படம் 9.4 Choosing the Slide Transition and Type

இந்தப் பக்கத்தில் உள்ள நிகழ்த்தல் சூழ்ச்சியம் (Presentation wizard) நிகழ்தலில் மாற்று விளைவுகளை உண்டு பண்ண உதவுகிறது. இந்தப் பக்கத்தின் மேற்பகுதியில் உள்ள Select a slide transition என்ற இணைப்பின் மூலம் ஒரு சில்லுக்கு வேண்டிய மாற்று விளைவு முறையையும், மாற்றத்தின் வேகத்தினையும் தேர்வு செய்து கொள்ளலாம். Select the presentation type, மாற்று விளைவு தானாக நிகழ வேண்டுமா ? அல்லது பயனாளர் கட்டுப்பாட்டில் இருக்க வேண்டுமா ? என்பதை தேர்வு செய்ய உதவுகிறது. Default பொத்தான் சுட்டெலியைக் கொண்டு கட்டுப்படுத்தும் வாய்ப்பை வழங்குகிறது. Automatic பொத்தான் ஒரு சில்லு எவ்வளவு நேரம் நிற்க வேண்டும் என்பதை உள்ளிடத் தூண்டுகிறது; மேலும் கொடுத்த நேரத்திற்குப்பின் சில்லு தானே மாறச் செய்கிறது.

Duration of Page சுழல்பெட்டி, சில்லுவின் நேரத்தைக் கொடுக்கவும், Duration of Pause சுழல்பெட்டி, நிகழ்த்தல் மறுபடியும் தொடங்குமுன் எவ்வளவு நேரம் இடை நிறுத்தம் (Pause) செய்ய வேண்டுமென்பதைக் கொடுக்கவும் உதவுகிறது.

Show Logo என்னும் இணைப்பு, இடை நிறுத்த நேரத்தில், காட்ட வேண்டிய செய்தியைக் காண்பிக்க உதவுகிறது.

முன்பக்கத்தில் உள்ளதைப் போல, Preview பெட்டி தேர்வு செய்த சில்லுவை முன்னதாகப் பார்க்க உதவுகிறது.

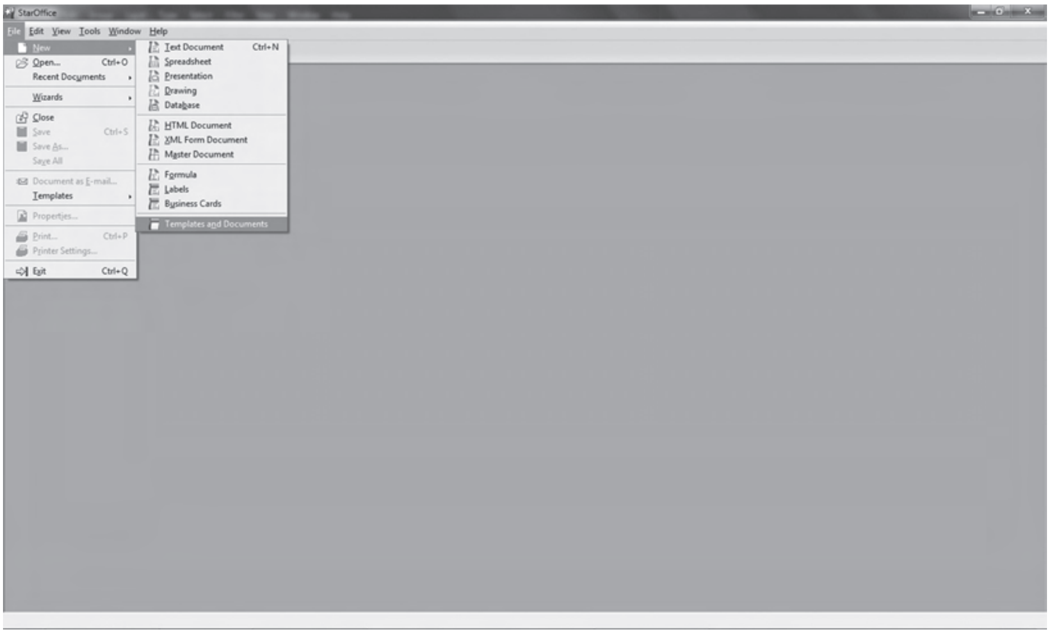
இப்பொழுது Create பொத்தானை கிளிக் செய்தால் நிகழ்த்தல் உருவாக்கப்படும். படம் **9.5–**இல் காட்டியுள்ளவாறு உருவாக்கிய சில்லு தோன்றும்.

படம் 9.5 StarOffice Impress Presentation Window

9.2.2 நிகழ்த்தல் கருவியின்றி நிகழ்த்தல் அமைத்தல் :

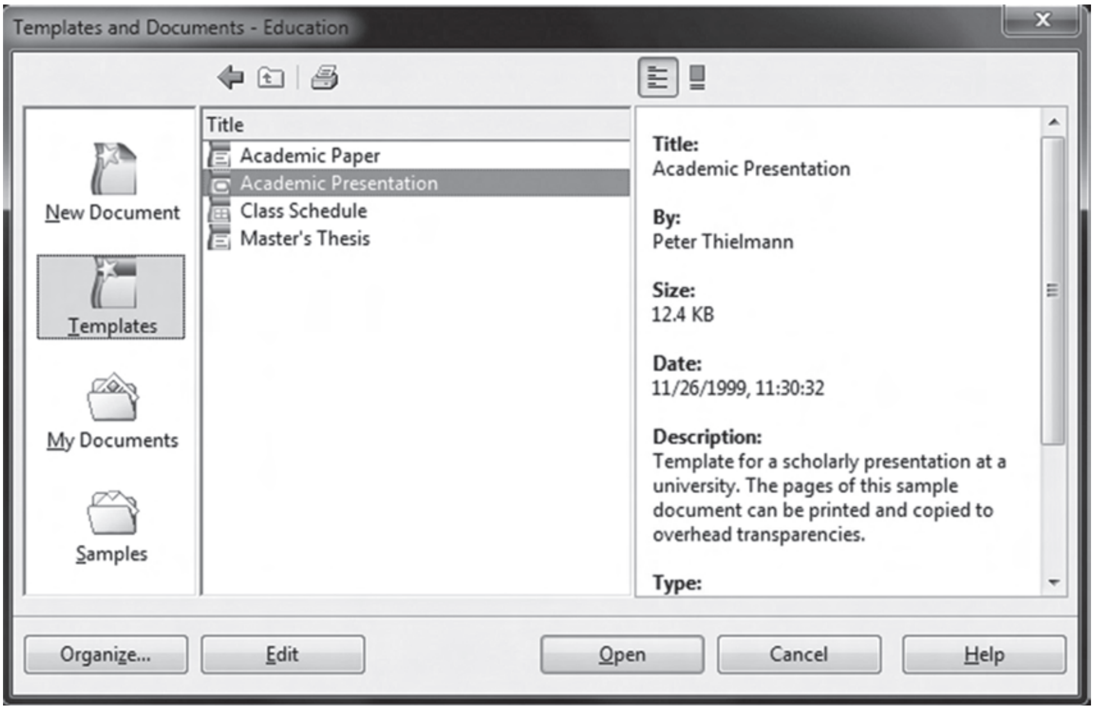
StarOffice Impress பல கொடாநிலை (Default) நிகழ்த்தல் படிவங்களைக் கொண்டுள்ளது. நிகழ்த்தல் உருவமைத்தலில் எளிதான முறை, இந்த கொடாநிலை நிகழ்த்தல் படிவங்களில் ஒன்றைத் தேர்வு செய்து, பின்னர் தேவைக்கேற்ப மாற்றிக் கொள்வதேயாகும். இதில் உள்ள நன்மை என்னவெனில், இந்த சில்லுகள் செய்தி வரையறைகளோடு உள்ளதால் விரைவில் வடிவமைக்க உதவுகிறது.

முன்வடிவமைத்த படிவங்களைக் (Template) கொண்டு நிகழ்த்தலை உருவமைக்க, File→ New→ Templates and Documents அல்லது Shif + Ctrl + N என்ற பொத்தான்களை கிளிக் செய்யவும். படம் **9.6 –**இல் காட்டியுள்ளவாறு சன்னல் ஒன்று தோன்றும்.



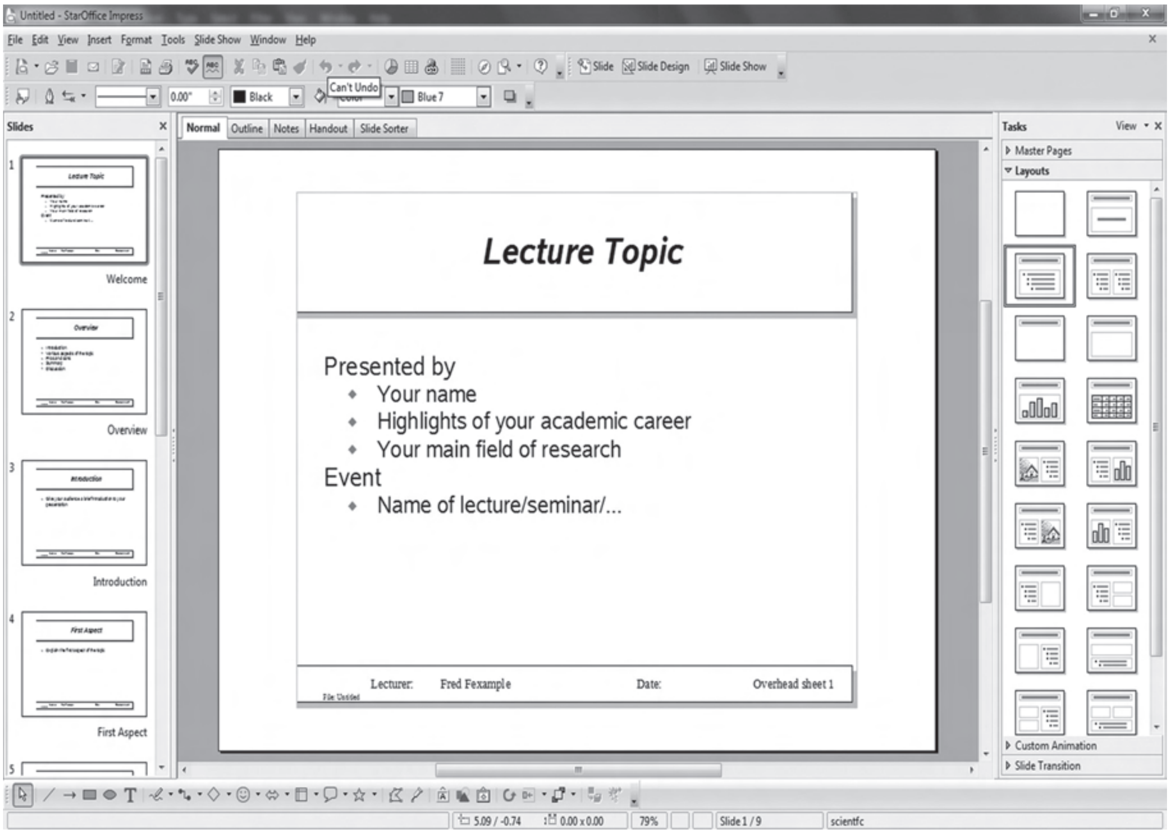
படம் 9.6 Choosing Templates and Documents Window

- அதில் Templates and Documents உரையாடல் பெட்டி உள்ளதைப் பார்க்கலாம். Templates பொத்தான் அழுத்தி Education பொத்தானை இருமுறை கிளிக் செய். பிறகு தோன்றும் முன்வடிவப் படிவங்களில் ஒன்றாகிய Academic Presentation பொத்தானை இருமுறை கிளிக் செய். படம் 9.7–இல் காட்டியுள்ள திரை தோன்றும்.



படம் 9.7 Choosing Academic Presentation from Templates Pane

9.2.3 நிகழ்த்தல் சன்னல் திரை



படம் 9.8 Selecting Academic Presentation from Templates Pane

படம் **9.8**–இல் காட்டியுள்ள Star Office Impress–இன் நிகழ்த்தல் சன்னல் திரையை சற்று கூர்ந்து நோக்குவோம். இதுவும் Star Office–இன் பிற பயன்பாட்டுத் திரையைப் போன்றதே. மேற்பகுதியில் ஒரு பட்டிப்பட்டை (Menu Bar) உள்ளது. அதற்குக் கீழே அடிக்கடி பயன்படும் கருவிகளின் குறுக்குவழி (Shortcut), குறும்படங்களை (icons), கொண்ட கருவிப்பட்டை (Toolbar) உள்ளது. அடியில் உள்ள நிலைமைப் பட்டி (Status bar) சில்லுகளின் எண்ணிக்கை போன்ற நிலைமைச் செய்தியைத் தெரிவிக்கின்றது.

திரையின் நடுவில் நிகழ்த்தலின் சில்லு இருப்பதைக் காணலாம். படத்தில் கண்டவாறு நடுப்பகுதியில் மூன்று சாளரப் பிரிவுகளைப் (Panes) பார்க்கலாம். இடப்புறத்தில் உள்ள சாளரப்பிரிவில் சிறு அளவில் சில்லுகள் உள்ளன. இப்பகுதியிலிருந்து சில்லுகளின் பெயர்களை மாற்றவோ, சில்லுகளை நீக்கவோ, சில்லுகளின் வரிசையை மாற்றி அமைக்கவோ செய்யலாம். நடுவில் உள்ள சாளரப் பகுதி சில்லுவைப் பார்க்கும் அளவிற்குக் காட்டுகிறது. இந்தச் சாளரப்பகுதியின் மேற்பகுதியில் ஐந்து தத்தல் (Tab) குறிகள் உள்ளன. இவற்றைக் கொண்டு நிகழ்த்தல் சில்லுவை பல்வேறு வடிவங்களில் பார்க்கலாம்.

Normal View	:	இது சில்லுவை உருவாக்கவும் பதிப்பிக்கவும் உதவுகிறது.
Outine View	:	இது சில்லுகளின் வரிசையை மாற்றி அமைக்கவும், சில்லுவின் பெயர்மற்றும் தலைப்புகளை தேவைக்கேற்ப பதிப்பித்துக் கொள்ளவும் உதவுகிறது.

Notes View : இது சில்லுகளில் குறிப்புகளைச் சேர்க்கவும் ஏற்கனவே உள்ள குறிப்புகளைப் பார்க்கவும் பயன்படுகிறது.

Handouts Views : இது சில்லுவின் அளவை மாற்றி பல சில்லுகள் ஒரு பக்கத்தில் வருமாறு செய்ய உதவுகிறது. இவ்வசதி சில்லுகளை அச்சிட்டு கைப்பிடி அறிக்கைகளாக (Handouts) வழங்க உதவுகிறது.

Slide Sorter View **:** இது எல்லா சில்லுகளையும் சிறிய வடிவில் ஒரே பக்கத்தில் பார்க்க உதவுகிறது.

இந்தச் சாளரப்பகுதியின் வலப்புறத்தில் உள்ள பணிச்சாளரத்தில் Master Page, Layouts, Custom Animation, Slide Transition ஆகிய கருவிகள், முதன்மை சில்லுவைக் குறிக்க, சில்லுவின் அமைப்பு மற்றும் மாற்று முறையை அமைக்க, அசைவுப்பட விளைவுகளை வழங்கப் பயன்படுகின்றன.

Master Page : இந்தப் பக்கம், எல்லா சில்லுகளிலும் வர வேண்டிய பின்னணிச் செய்திகளைச் சேர்க்க உதவுகிறது. எடுத்துக்காட்டாக, ஒரு நிறுவனத்தின் முத்திரைச் சின்னம் எல்லா சில்லுகளிலும் வர வேண்டுமெனில், அதனை Master Page–இல் கொடுத்துவிட்டால், அது எல்லா சில்லுகளிலும் தோன்றச் செய்யும்.

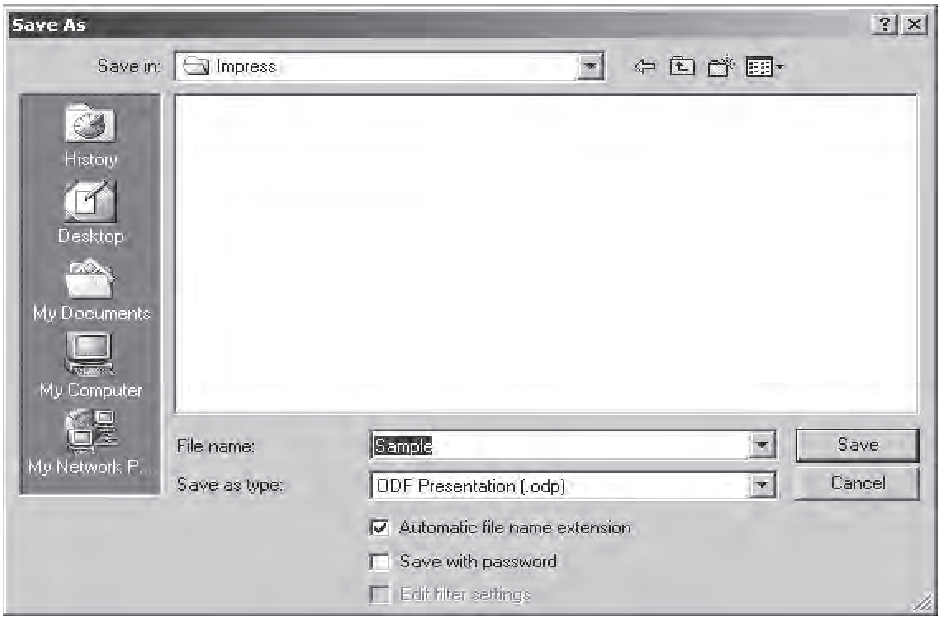
Layouts **:** இந்தப் பக்கம், Star Office Impress வழங்கும் அனைத்து சில்லு அமைப்புகளையும் காட்டும். புது சில்லுகள் உருவாக்கும் போது அதில் எதையேனும் தேர்வு செய்து கொள்ளலாம்.

Custom Animation : இந்தப்பக்கம், சில்லுவில் வழங்கும் பொருள்களுக்கு அசைவுப்பட விளைவுகளை சேர்க்கவோ, மாற்றி அமைக்கவோ உதவுகிறது.

Slide Transition **:** இந்தப் பக்கம், சில்லுகளுக்கு மாற்று விளைவுகளை வழங்குவதற்கான பல மாற்று விளைவு முறைகளைக் கொண்டுள்ளது. ஒவ்வொரு சில்லுக்கும் வெவ்வேறு மாற்று விளைவுகளை வழங்கலாம் என்பதைக் கவனிக்கவும்.

9.2.4 நிகழ்த்தலைச் சேமித்தல்

ஒரு நிகழ்த்தல் உருவமைப்பை சேமிக்க File→Save பொத்தான்கள் அல்லது என்ற குறும்படத்தை கிளிக் செய்யவும். இந்த நிகழ்த்தல் முன்பே சேமிக்கப்படவில்லை எனில் படம் **9.9** –இல் காட்டியுள்ளSave As உரையாடல் பெட்டி தோன்றும். File name text box –இல் நிகழ்த்தலின் பெயரை அச்சிட்டு Save என்ற பொத்தானை கிளிக் செய்யவும்.



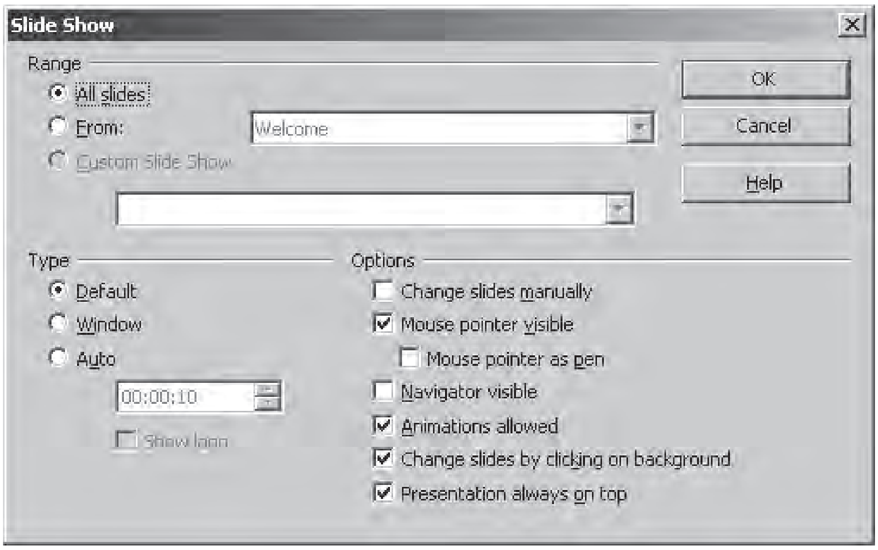
படம் **9.9 Save as dialog box**

9.3 நிகழ்த்தலை கையாளல்

9.3.1 நிகழ்த்தலை தொடங்குதல்

ஒரு நிகழ்த்தலைத் தொடங்க, நிகழ்த்தல் கருவிப்பட்டையில் உள்ள slideshow குறும்படம் அல்லது Slide Show → Slide show அல்லது F5 பொத்தானை கிளிக் செய்யவும் நிகழ்த்தல் முழுத்திரையில் தோன்றும். பிறகு கிளிக் செய்யச் செய்ய சில்லுகள் அடுத்து தோன்றும். ஒவ்வொரு முறையும் சில்லுகள் ஒவ்வொன்றும் அவற்றுக்குக் கொடுக்கப்பட்டுள்ள மாற்று விளைவுகளுடன் தோன்றும். இறுதியில் கருப்புத் திரை தோன்றும். நிகழ்த்தலை முடிக்க Escape சாவிப் பொத்தானை அழுத்தவும்.

Slide Show → Slide Show Settings –இல் உள்ள வசதிகளைப் பயன்படுத்தி நிகழ்த்தல் அமைவுகளை (Presentation Settings) சரி செய்து கொள்ளலாம். (படம்– 9.10).



படம் **9.10 Slide Show Settings**

எல்லா சில்லுகளையும் பார்க்க All Slides இணைப்பைத் தேர்வு செய்க. இன்றேல் From இணைப்பைத் தேர்வு செய்து, எந்தப் பக்கத்திலிருந்து பார்க்க வேண்டுமோ அந்தப் பக்கத்தைத் தேர்வு செய்க.

சில்லுகளைப் பார்க்க நாம் விரும்பும் முறையை Slide Show →custom slide show என்ற இணைப்புகளைக் கிளிக் செய்து கொடுத்த பின், அதனைப் பார்க்க CustomSlide Show இணைப்பைத் தேர்வு செய்க.

9.3.2 சில்லுகளை சேர்த்தல், நீக்குதல், மாற்றுப் பெயரிடல்

ஒரு சில்லை நிகழ்த்தல் தொகுப்பில் சேர்க்க நிகழ்த்தல் கருவிப்பட்டியில் உள்ள slide பொத்தானை கிளிக் செய்யலாம் Insert →Slide அல்லது இணைப்புகளை பட்டிப்பட்டை (Menu Bar) யிலிருந்து தேர்வு செய்யலாம். புது சில்லு முன் சில்லுவின் பக்க அமைப்பைப் பயன்படுத்தி உருவாக்கப்படும்.

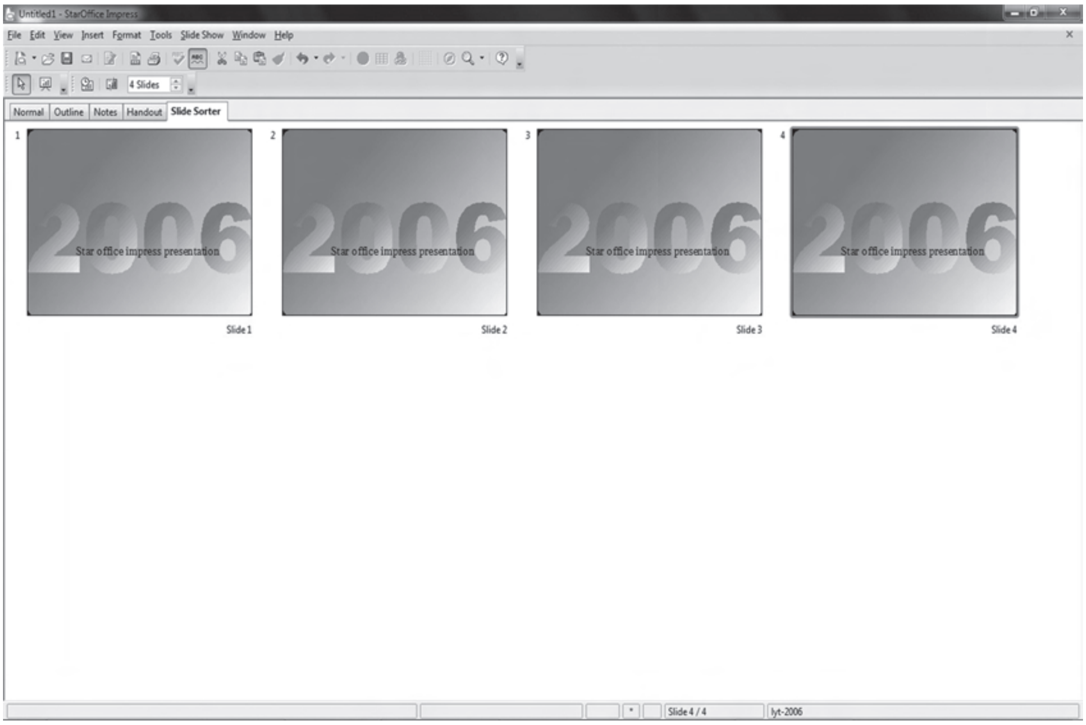
ஒரு சில்லை நீக்க வேண்டுமானால், நீக்க வேண்டிய சில்லுவைத் தேர்வு செய்து Delete சாவிப் பொத்தானை அழுத்தவும் அல்லது சுட்டெலியின் வலப்புற பொத்தானை கிளிக் செய்து Delete Slide–ஐத் தேர்வு செய்யவும்.

ஒரு சில்லுவின் பெயரை மாற்றி மறுபெயர் இட, சில்லுவைத் தேர்வு செய்து Slide→ Rename Slide இணைப்பைத் தேர்வு செய்க. அல்லது சுட்டெலியின் வலப்புற பொத்தானை கிளிக் செய்து Rename Slide–ஐத் தேர்வு செய்க. (படம் 9.11).

படம் 9.11 Renaming a Slide

9.3.3 சில்லுவின் வரிசைமுறை மாற்றல்

சில்லுவின் வரிசை முறை மாற்ற Switching Presentation View தத்தலில் உள்ள Slide Sorter பொத்தான் மீது கிளிக் செய்க. படம் 9.12 காட்டியுள்ள சன்னல் திரை தோன்றும். இதில் எல்லா சில்லுகளும் சிறிய அளவில் காட்டப்படும். நகர்த்த வேண்டிய சில்லு மீது கிளிக் செய்து சுட்டெலியைக் கொண்டு விரும்பிய இடத்திற்கு நகர்த்தவும். ஒரு கருப்பு நெடுங்கோடு சில்லு நகர்த்தப்பட வேண்டிய இடத்தைச் சுட்டிக் காட்டும்.



படம் 9.12 Slide Sorter window

9.3.4 படங்கள், பொருள்கள், ஒலி, ஒளிக்காட்சிகள் செருகல்

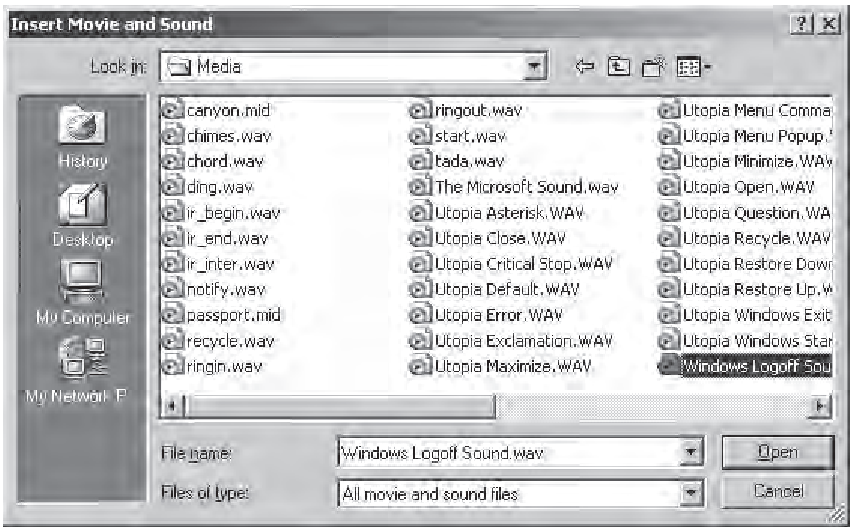
படங்களைச் செருகுதல்

ஒரு சில்லுவில் படம் ஒன்றை செருக Insert → Picture→ From File இணைப்புகளைத் தேர்வு செய்க. அல்லது Picture குறும்படக்குறி –ஐ Insert கருவிப்பட்டையிலிருந்து தேர்வு செய்து கிளிக் செய்க. ஒரு உரையாடல் பெட்டி தோன்றும். அதிலிருந்து சேர்க்க விரும்பும் படத்தை தேர்வு செய்க. படம் 9.13–இல் காட்டியுள்ளவாறு தேர்வு செய்யப்பட்ட படம் சேர்க்கப்படும்.

படம் 9.13 Inserting a Picture

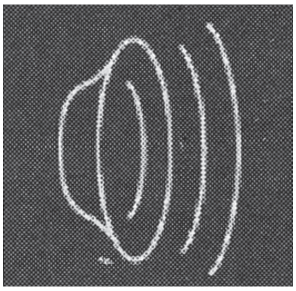
ஒலி, ஒளிக்காட்சிகள் சேர்க்கை

ஒரு ஒலி அல்லது ஒளிக்காட்சியைச் செருக, Insert → Movie and Sound இணைப்புகளைத் தேர்வு செய்க. அல்லது ஒலி–ஒளிச் சேர்க்கை குறும்படத்தை கிளிக் செய்க. படம் **9.14**–இல் காட்டியுள்ள ஒலி– ஒளி உரையாடல் பெட்டி தோன்றும். செருகப்பட்ட ஒலிக்கோப்பு சில்லுவில் படம்**9.15**–இல் உள்ளவாறு தோன்றும்.



படம் 9.14 Insert Movie and Sound dialog box

ஒரு ஒளிக்காட்சி அல்லது ஒலிக்கோப்பை ஓட்டிப் பார்க்க கீழ்க்காணுமாறு செய்க:



படம் 9.15 Sound file display in slide

1. ஒளிக்காட்சி அல்லது ஒலிக்கோப்புள்ள சில்லுவைத் திற.
2. சில்லுவில் உள்ள ஒலி அல்லது ஒளிக்காட்சி குறும்படத்தின் மீது கிளிக் செய்.
3. Media Playback கருவிப் பட்டையில் உள்ள Play பொத்தானை கிளிக் செய்க.

Media Playback கருவிப் பட்டை படம் **9.16**–இல் காட்டப்பட்டுள்ளது. இதில் இடைநிறுத்தல், நிறுத்துதல், இணைத்தல் ஒலி அளவைக் கட்டுப்படுத்தல்

படம் 9.16 Media Playback Toolbar

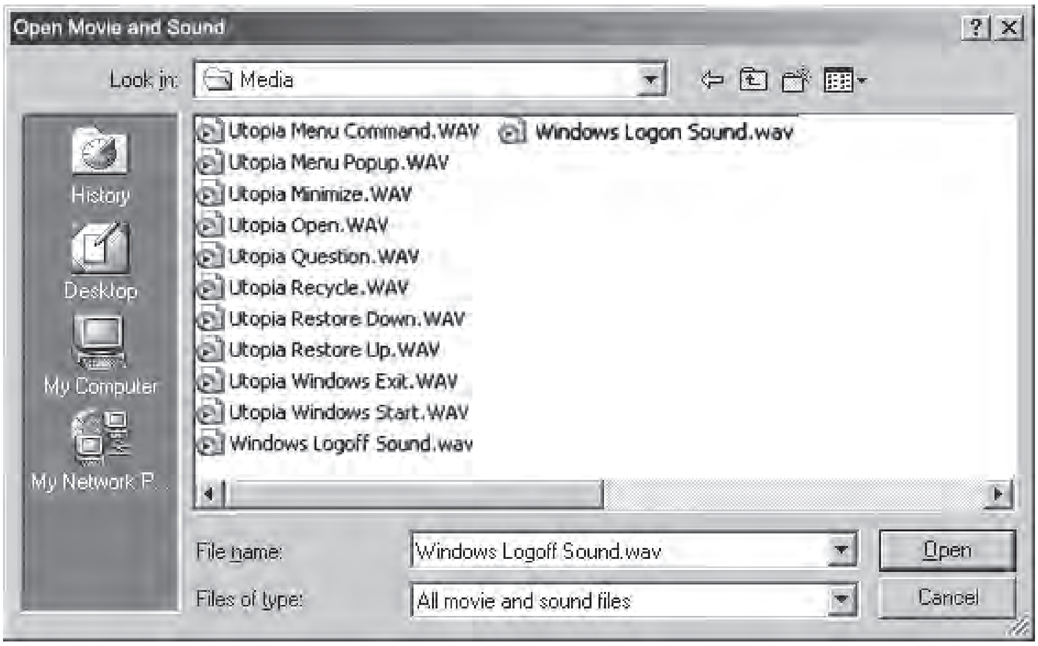
ஒலியை இடைநிறுத்தம் செய்தல் போன்ற பல வசதிகள் உள்ளன. இடப்புற நகர்வி (Slider) எந்த அளவு ஓடியுள்ளது என்ற நிலைமையைக் காட்டுகிறது. வலப்புற நகர்வி ஒலி அளவைக் கட்டுப்படுத்த உதவுகிறது. ஒளிக்காட்சி கோப்புகளுக்கு படத்தைப் பெரிது படுத்தும் பட்டியல் பெட்டியும் உள்ளது.

	Play icon	-	நிகழ்கோப்பை ஓட்டுகிறது.
	Pause icon	-	நிகழ் கோப்பை இடைநிறுத்தம் செய்யவும் திருப்பி ஓட்டவும் உதவுகிறது.
	Stop icon	-	நிகழ் கோப்பை நிறுத்துகிறது.
	Repeat icon	-	கோப்பை திருப்பித் திருப்பி ஓட்ட உதவுகிறது
	Position slider	-	கோப்பின் பல்வேறு பகுதிகளுக்குச் செல்ல உதவுகிறது.
	Mute icon	-	ஒலியை நிறுத்தவும் தொடங்கவும் உதவுகிறது.
	Volume slider	-	ஒலி அளவைக் கட்டுப்படுத்துகிறது.
	Zoom	-	ஒளிக்காட்சியின் அளவைக் கட்டுப்படுத்த உதவுகிறது.

Star Office Impress, Media Player மூலம் ஒலி, ஒளிக்காட்சி கோப்புகளை முன்னதாகப் பார்க்க வழி செய்கிறது. Media Player பல ஊடகப் படிவங்களை வழங்க உதவுகிறது. Media Player சன்னல் திரையை திறக்கTools→Media Player–ஐத் தேர்வு செய்க. படம் 9.17–இல் உள்ளதைப்போல Media Player சன்னல் திரை தோன்றும்.

படம் 9.17 Media Player

Open குறும்படத்தைக் கிளிக் செய்தால், படம் 9.18–இல் உள்ளதைப் போல ஒலி–ஒளிக்காட்சி உரையாடல் பெட்டி தோன்றும். ஒரு ஒலி அல்லது ஒளிக்காட்சிக் கோப்பைத் திறந்து முன்பார்வை செய்க. பிறகு Apply குறும்படத்தை கிளிக் செய்து நிகழ்த்தலில் சேர்த்துக்கொள்க. பிறகு Media Player சன்னல் திரையை மூடி விடுக.



படம் 9.18 Open Movie and Sound dialog box

Slide Show→Slide Show–ஐ தேர்வு செய்து அல்லது சாவி F5–ஐ அழுத்தி சில்லு காட்சியைத் தொடங்குக. படம் 9.19–இல் ஒரு சில்லுக்காட்சியில் ஒரு ஒளிப்படக்காட்சி குறிப்பை இட்டுள்ளதைப் பார்க்கலாம்.

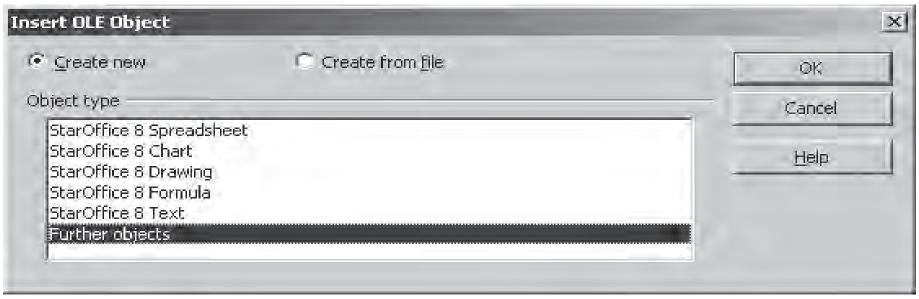
படம் 9.19 Movie file played during slide show

பொருட்குறிப்பு (Object) சேர்த்தல்

Star Office impress நிகழ்த்தலில் வரைபடம், வாய்பாடு போன்றவற்றை இடுவதற்கான கருவிகளை வழங்குகிறது. ஒரு சில்லுவில் ஒரு பொருட்குறிப்பைச் செருக Insert → Object– லிருந்து ஒரு பொருளைத் தேர்வு செய்க.

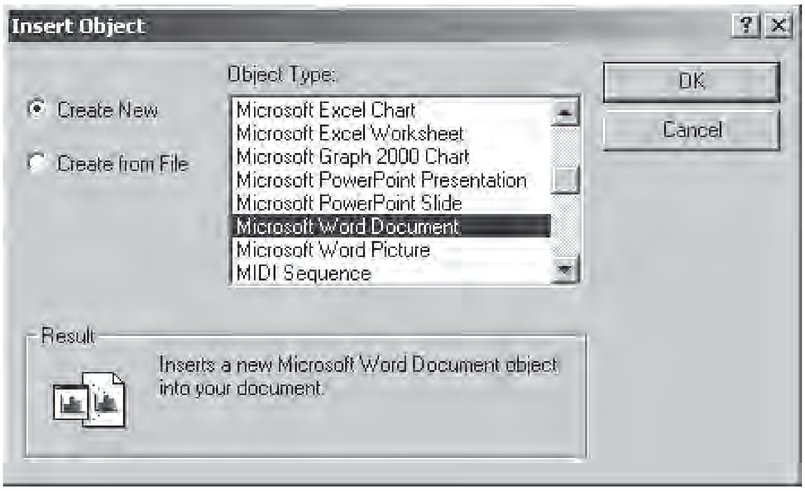
Insert Fromula : இது கணக்கீடுகளைச் செய்வதற்கான வாய்பாட்டினை நிகழ்த்தலில் செருக உதவுகிறது.

Insert OLE object : இது பிற பயன்பாடுகளிலிருந்து பொருள்களை நிகழ்த்தலுக்குள் தருவிக்க உதவுகிறது. எடுத்துக்காட்டாக, Star Office அட்டவணைப் பக்கம் (Spread sheet), வாய்பாடு. உரை, Microsoft–இன் பயன்பாட்டுப் பொருட்கூறுகள் உட்பட எல்லாவற்றையும் நிகழ்த்தலில் கொணர்ந்து சேர்க்கலாம். படம் **9.20**, படம் **9.21** ஆகியவற்றில் Microsoft–இன் சொல் ஆவணத்தை (Word Document) கொணர்ந்து சேர்ந்துள்ளமை காட்டப்பட்டுள்ளது.



படம் 9.20 Inserting OLE Object

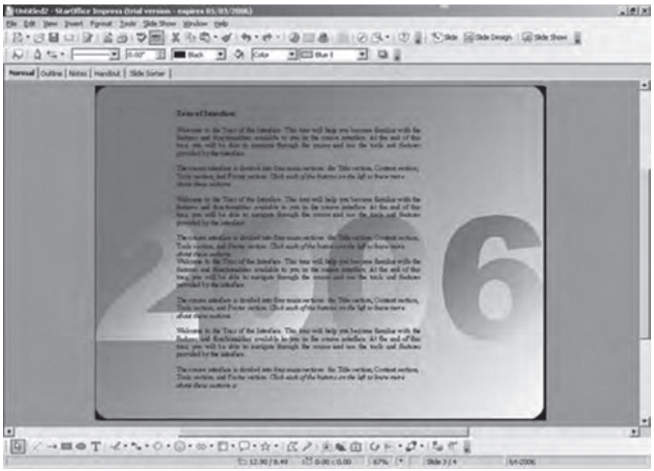
Microsoft சொல் ஆவணத்தை கொணர்ந்து சேர்க்க, Further objects, creative New ஆகியவற்றை Insert OLE object உரையாடல் பெட்டியிலிருந்து தேர்வு செய்து கிளிக் செய்க. Insert object உரையாடல் பெட்டியிலிருந்து Microsoft Document–ஐ பொருள் வகையாகத் தேர்வு செய்து கிளிக் செய்க.



படம் 9.21 Object Type Selection window

Microsoft Word பயன்பாடு தானே திறக்கும். பிறகு, Ms word –இல் செய்தியை அச்சிட்டு மூடுக. Star Office Impress தானாக ஆவணத்தை சேமித்து சில்லுக்குள் இடுகிறது. படம் **9.22** செருகப்பட்ட Ms word ஆவணத்தை காட்டுகிறது.

Insert Applet : இது செய்நிரலை (Applet) நிகழ்த்தலில் செருக உதவுகிறது.



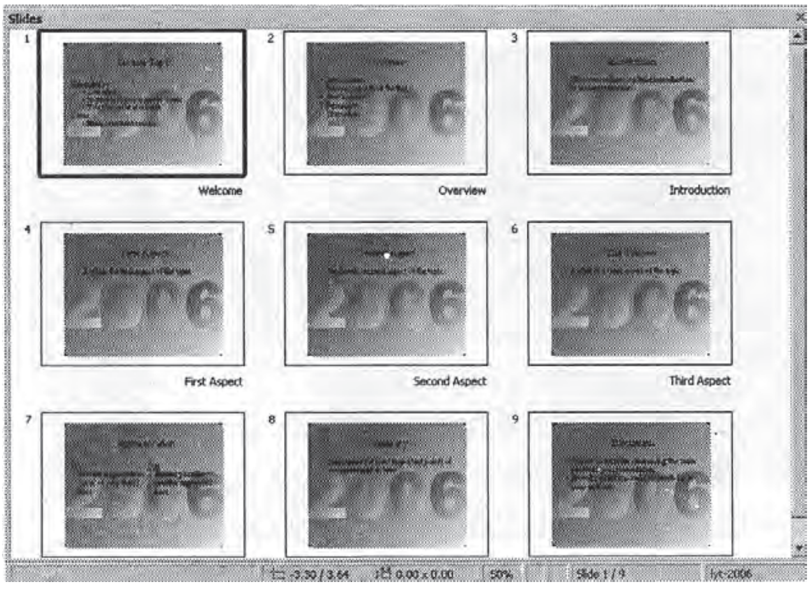
படம் 9.22 OLE Object in Presentation

9.3.5 சில்லு மாற்று முறை, விளைவுகளும் அசைவுப்படமும்

9.3.5.1 தானியங்கு சில்லு மாற்றுமுறை

சில்லு மாற்று விளைவுகளை அமைக்கும் எளிய முறை Slide View–இல் உள்ளது. View→Tools bar →Slide View ஆகியவற்றைக் தேர்வு செய்க. பிறகு Slides Per Row சுழற்பெட்டியில் பட்டியலிட வேண்டிய சில்லுகளின் எண்ணிக்கையைத் தேர்வு செய்க. படம் 9.23–இல் காட்டியுள்ளவாறு, Slides Per Row சுழற்பெட்டியில் தேர்வு செய்த எண் தோன்றும். தேர்வு செய்த சில்லுகளைSlides View படம் 9.24 உள்ளவாறு காட்டுகிறது.

படம் 9.23 Slide View Floating window



படம் 9.24 Slide View window

Effects for Objects :

சில்லுவில் உள்ள பொருள்கள் பல்வேறு விளைவுகளைக் கொண்டிருக்கலாம். எடுத்துக்காட்டாக, அவை நிகழ்த்தலில் இடமிருந்தோ, வலமிருந்தோ நகரலாம். உரை மெதுவாகத் திரையில் தோன்றலாம். இப்படிப் பல விளைவுகளை வழங்கலாம்.

இவ்விளைவுகளையெல்லாம் Normal View (View→Normal View) –ஐப் பயன்படுத்தி உண்டாக்கலாம்.

Slide Show→Custom Animation –லிருந்து Custom Animation–ஐ தேர்வு செய்க. முதலில் எந்தப்பொருளுக்கு அசைவுப்பட விளைவு கொடுக்க வேண்டுமோ அந்தப் பொருளைத் தேர்வு செய்க. பிறகுModify Effect–லிருந்து Add பொத்தானை கிளிக் செய்க. படம் 9.25–இல் காட்டியுள்ளவாறு அசைவுப்பட அமைப்பு சன்னல் திரை தோன்றும்.

படம் 9.25 Custom Animation window

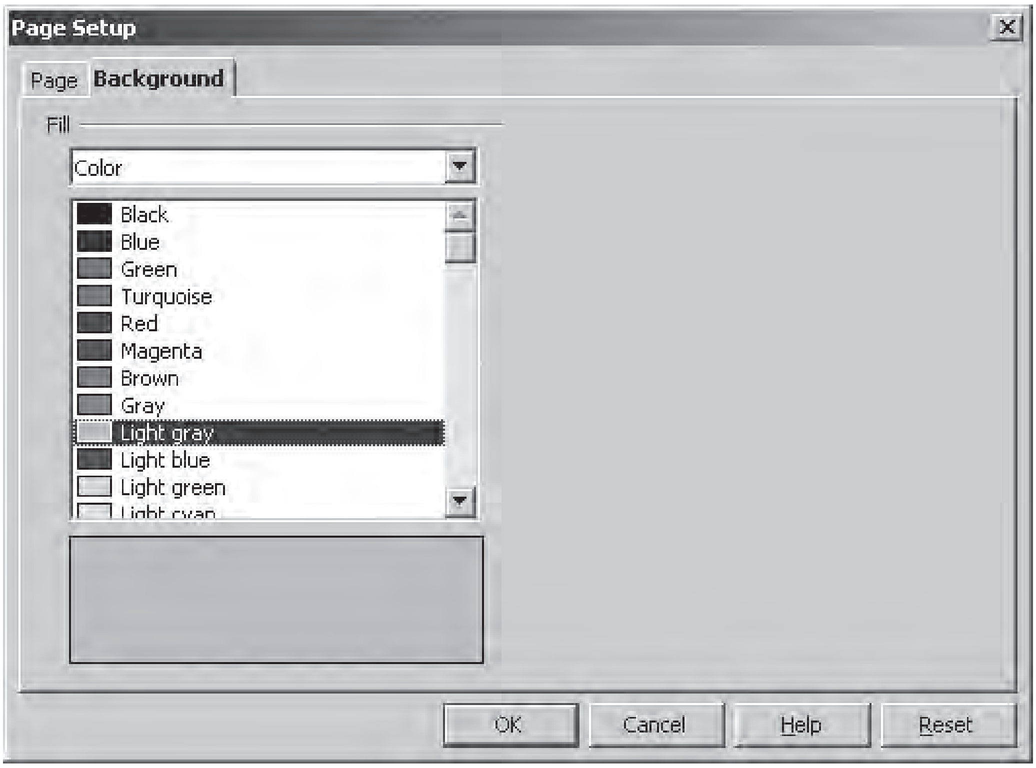
பிறகு தேவையான விளைவைத் தேர்வு செய்து OK பொத்தானை கிளிக் செய்யவும். Custom Animation சன்னல் திரையில் Special வகைக்கீழ் Entrance பொத்தானை கிளிக் செய்து காட்சிப் பொருளை மங்கலாகச் செய்யலாம். Exit பொத்தானை கிளிக் செய்து வெளியேறலாம்.

சில்லுவின் பின்னணியை மாற்றல்

ஆவணத்தின் அப்பொழுது திரையில் உள்ள அல்லது எல்லா சில்லுகளின் பின்னணி மற்றும் பின்னணியின் நிரப்பல் ஆகியவற்றை மாற்றலாம். பின்னணி நிரப்பலுக்கு hatching, gradient அல்லது bitmap நிழற்படம் ஆகியவற்றைப் பயன்படுத்தலாம். **(படம் 9.26)**

படம் 9.26 Background Fill combo box

Format→Page→Background ஆகிய இணைப்புகளைத் தேர்வு செய்து, Color, Gradient, Hatching, Bitmap Image ஆகியவற்றில் ஒன்றைத் தேர்வு செய்க. எடுத்துக்காட்டாக, Fill வகையில் Color–ஐ தேர்வு செய்து கிடைத்த திரையில் பல்வேறு வண்ண வகைகள் காட்டப்படுவதை படம் 9.27–இல் பார்க்கலாம்.



படம் 9.27 Selecting Background Color

Fill பகுதியில் கீழ்க்காணும் வகைகளில் ஒன்றைத் தேர்வு செய்க:

- Color தேர்வு செய்து, பட்டியலில் ஒரு வண்ணத்தை கிளிக் செய்க.
- Gradient தேர்வு செய்து, பட்டியலிலிருந்து ஒரு சாய்வளவைக் கிளிக் செய்க.
- Hatching தேர்வு செய்து ஒரு வரிவேய் முறையை கிளிக் செய்க.
- Bitmap தேர்வு செய்து பட்டியலிருந்து ஒரு நிழற்படத்தை கிளிக் செய்க.

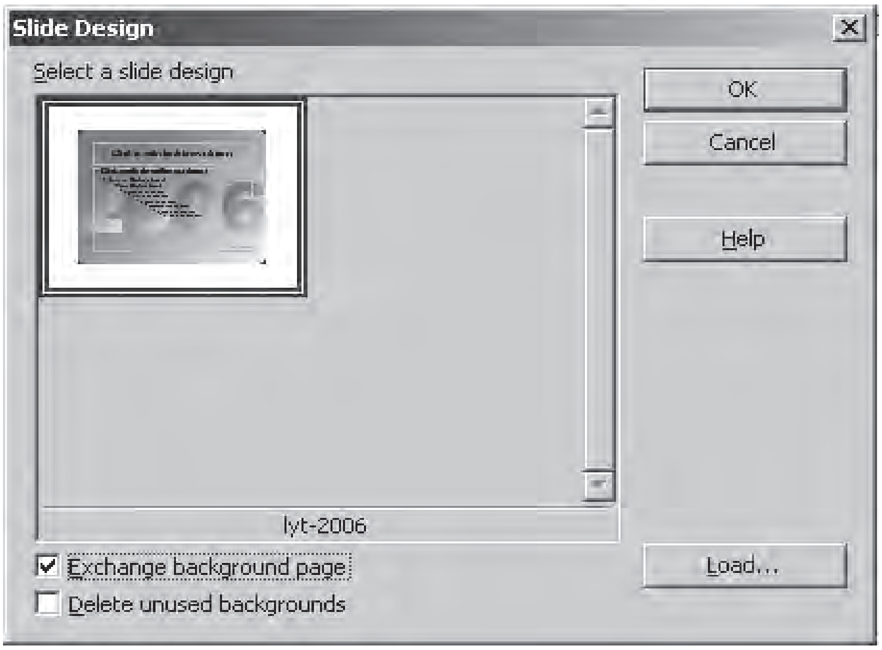
பிறகு OK பொத்தானை கிளிக் செய்க. Page Settings உரையாடல் பெட்டி தோன்றும் (படம் 9.28). எல்லா சில்லுகளின் பின்னணி நிரப்பல்களையும் மாற்ற வேண்டுமானால் Yes பொத்தானை கிளிக் செய்யவும். ஒன்றை மட்டும் மாற்ற விரும்பினால் No பொத்தானை கிளிக் செய்யவும்.

படம் 9.28 Background Fill Dialog Box

ஒரு முதன்மைச் சில்லுவில் சில்லுவை வடிவமைத்தல்

ஒரு முதன்மை சில்லு, ஒன்று அல்லது அனைத்து சில்லுகளின் தலைப்பு வடிவமைப்பு வகை. வரைச்சட்டம் **(outline),** பின்னணி வடிவமைப்பு ஆகியவற்றை முடிவு செய்கிறது.

சில்லுவை வடிவமைக்க, **Format→Slide design** தேர்வு செய்க. படம் 9.29–இல் உள்ளவாறு உரையாடல் பெட்டி தோன்றும். பிறகு கீழ்க்காணும் ஏதேனும் ஒன்றைச் செய்யவும்.

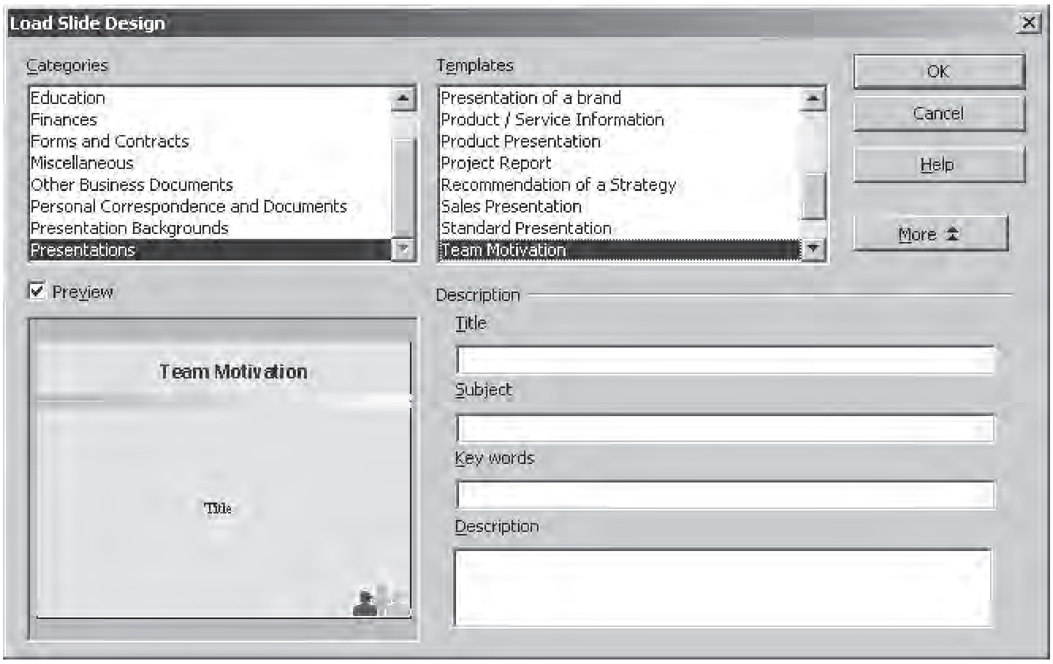


படம் 9.29 Slide Design window

எல்லா சில்லுகளுக்கும் சில்லு வடிவமைப்பு செய்ய வேண்டுமெனில் Exchange Background Page பெட்டியை தேர்வு செய்க.

அப்பொழுது திரையில் உள்ள ஒரு சில்லுவை மட்டும் வடிவமைக்க விரும்பினால்Exchange Background Page–ஐ துடைத்து (clear) விடவும்.

Load பொத்தானை கிளிக் செய்க. படம் 9.30–இல் உள்ள சன்னல் திரை தோன்றும் Categories–கீழ்Slide design–ஐத் தேர்வு செய்க.



படம் 9.30 Load Style Design window

Template கீழ் ஒரு முன்வடிவைத் (a template) தேர்வு செய். முன் பார்க்க வேண்டுமெனில் Move பொத்தானை கிளிக் செய்து Preview பொத்தானை அழுத்தவும். பிறகு OK பொத்தனை கிளிக் செய்து முதன்மை சில்லுவில் ஏற்படுத்தியுள்ள மாற்றங்களைப் பார்க்கலாம்.

நிகழ்த்தல் பாணிகள்

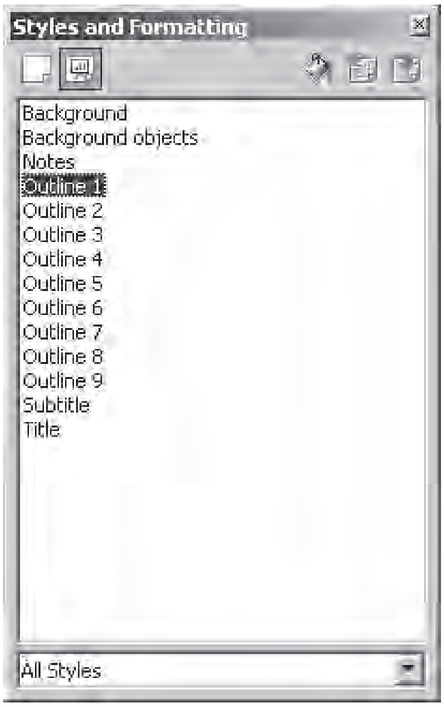
சில்லு வடிவமைப்புடன், சில்லுகளுக்கு நிகழ்த்தல் பாணிகளையும் (Presentation Styles) வழங்கலாம்.Style list பெட்டியைத் திறந்து வேண்டிய பாணியைத் தேர்வு செய்து கொள்ளலாம். அப்பொழுது திரையில் உள்ள சில்லுவின் பாணியை மாற்றலாம் அல்லது புது பாணியை வழங்கலாம். Outline 1 முதல் Outline 9 வரை உள்ள பாணிகள் சில்லுகளின் பெயர்களையும் தலைப்புகளையும் ஒரு சீராக அமைக்க உதவுகின்றன.

ஒரு புது, காலி நிகழ்த்தலை ஒரு சில்லு இட அமைவுடன் (Layout) வரைச்சட்ட நிலைகளை (Outline levels) தேர்வு செய்யும் வகையில் திறந்து கொள்ளவும். “Title Text” என்னும் சில்லு இட அமைவு (Layout) இதைச் செய்ய உதவுகிறது.

குறிப்பு : தேர்வு செய்யும் சில்லு இட அமைவு (Slide loyout) வரைவுச் சட்டங்களின் நிலைகளை (Outline levels) முடிவு செய்கிறது. தேர்வு செய்யும் சில்லு இட அமைவு Outline வசதியை வழங்கவில்லை எனில், Outline வடிவமைப்பிற்குள் செல்ல இயலாது.

View→Outline View இணைப்புகளைத் தேர்வு செய்து Outline தேர்வை செயல்படுத்தலாம்.

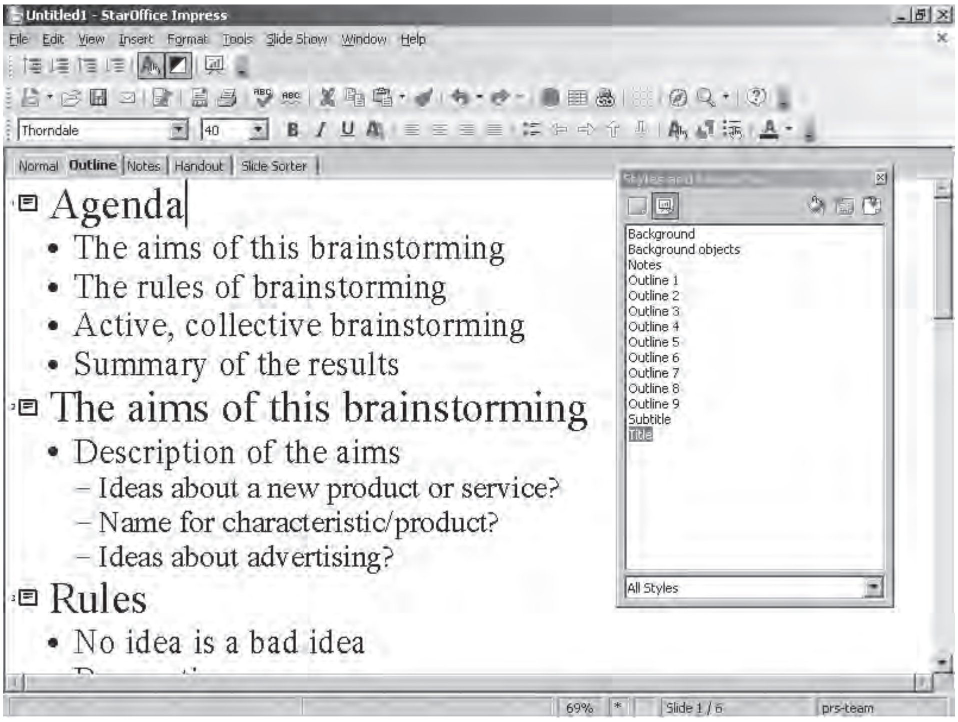
Style list–ஐ, Format→Styles and Formatting அல்லது F11 சாவியை தேர்வு செய்து திறக்கவும். படம் 9.31–இல் உள்ள திரை தோன்றும்



படம் 9.31 Styles and Formatting window

- சில்லு 1–க்கு உரிய சிறு குறும்படத்திற்கு அருகில் சுட்டி இருக்கிறது. இங்கே விரும்பிய உரையை அச்சிடவும். எடுத்துக்காட்டாக 'First Page' எனக் கொடுக்கவும். இது சில்லு–1 இன் தலைப்பாக அமையும்.
- Enter சாவியை அழுத்தினால் Outline View–வின் அடுத்த வரியில் சில்லு –2க்கான சிறு குறும்படம் தோன்றும். இங்கே இரண்டாவது சில்லுவின் தலைப்பைக் கொடுக்கவும். எடுத்துக்காட்டாக Second Page என்று கொடுக்கவும்.
- Enter சாவியை மீண்டும் அழுத்தவும். இப்பொழுது வேறு உரை ஏதும் கொடுப்பதற்கு முன் Tab சாவியை அழுத்தவும். இப்படிச் செய்வதால், இப்பொழுது கொடுக்கப்போகும் உரை இரண்டாவது சில்லுவின் துணைத் தலைப்பாக அமையும்.
- வேண்டுமானால் துணைத் தலைப்பிற்கு உரிய உரையை முதலிலேயே அச்சிட்டு Tab சாவியை அழுத்தலாம். அப்படிச் செய்யும்போது, "இந்தச் செய்கையால் நீ துடைத்துவிடுவாய்" என்ற எச்சரிக்கை செய்தி திரையில் தோன்றும். துடைக்க OK பொத்தானை Click செய்க.
- பக்கம் இரண்டின் முதல் நிலையில் மேற்கொண்ட துணைத் தலைப்புகளை உள்ளிடவும். Tab சாவியை அழுத்தி நிலைக்குக் கீழேயும், Shift + Tab சாவிகளை அழுத்தி நிலைக்கு மேலேயும் செல்லலாம். இது துணைத் தலைப்புகளிலிருந்து புது சில்லுகளைச் செய்யவும் இது வழி செய்கிறது.

மேற்கண்டவாறு பல நிலைகளை உருவமைத்த செயல், தானாகவே நிகழ்த்தல் சில்லுகளின் தலைப்பில் Outline 1, Outline 2… என உருவமைக்கப்படுகிறது. புது சில்லு வடிவமைப்பை இவற்றிற்குள் வழங்கி இந்தப் பாணிகளை மாற்றி அமைக்கலாம். எடுத்துக்காட்டாக, புது எழுத்துரு, எழுத்து அளவு, எழுத்துக்களின் வண்ணம் ஆகியவற்றை அமைக்கலாம். இந்த வடிவமைப்புகளை பதிப்பு செய்வதென்றாலும் செய்யலாம். Outline View படம் 9.32–இல் காட்டப்பட்டுள்ளது.



படம் 9.32 Outline View of Presentation

9.4 நிகழ்த்தலை தனிப்பயனாக்கல் (Customizing a Presentation)

தனிப்பயனுள்ள பல சில்லுக்காட்சிகளை உருவாக்கலாம். **Star Office Impress** இப்பொழுது திரையில் உள்ள சில்லுவிலிருந்து சில்லுக்காட்சியைத் தொடங்கலாம். ஒரு சில்லுக் காட்சியின்போது சில்லுகளை மறைக்கவும் செய்யலாம்.

தனிப்பயன் சில்லுக்காட்சி அமைத்தல்

- Slide Show → Custom Slide Show – ஐத் தேர்வு செய்து New பொத்தானைக் கிளிக் செய்க.
- படங்கள் 9.33, 9.34 ஆகியவை Custom Slide Show – இன் உரையாடல் பெட்டியைக் காட்டுகின்றன.
- Name பெட்டியில் சில்லுக்காட்சிக்கான பெயரைக் கொடுக்கவும்.
- இருக்கின்ற சில்லுகளிலிருந்து சில்லுக்காட்சியில் சேர்க்க விரும்பும் சில்லுகளை தேர்வு செய்து >> பொத்தானை கிளிக் செய்க. Shift பொத்தானை அழுத்தி சில்லுகளைத் தேர்வு செய்து கொள்ளலாம். Ctrl சாவியை அழுத்தி ஒரே நேரத்தில் ஒன்றுக்கு மேற்பட்ட சில்லுகளைத் தேர்வு செய்து கொள்ளலாம்.

- தனிப்பயன் சில்லுக்காட்சியில் சில்லுகளின் வரிசையை மாற்றிக் கொள்ளலாம். இதனை Selected Slide – இல் சென்று சில்லுகளை பிடித்து இழுத்து நகர்த்தி வேண்டிய இடத்தில் இட்டுக் கொள்ளலாம்.
- இப்பொழுது OK பொத்தானை கிளிக் செய்க.

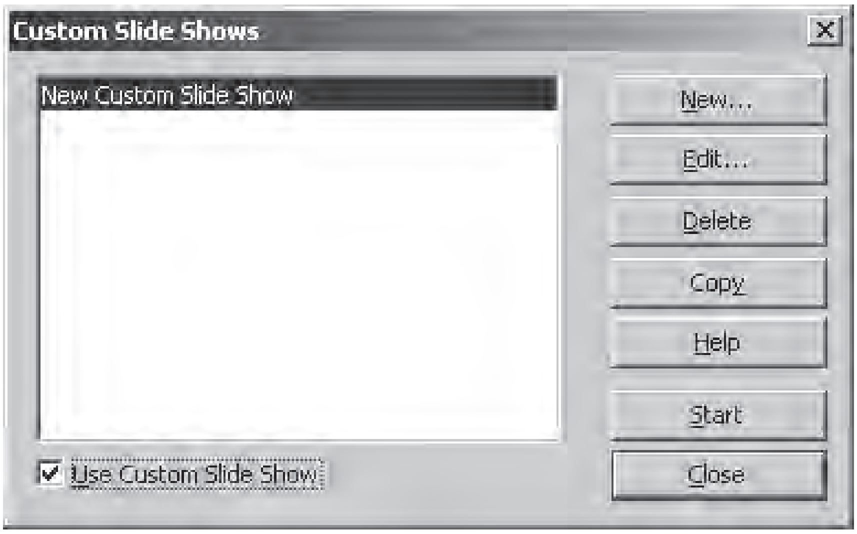
படம் 9.33 Creating New Custom Slide show

படம் 9.34 Define Custom Slide show

தனிப்பயன் சில்லுக்காட்சியைத் தொடங்க

- **Slide Show→Custom Slide Show** தேர்வு செய்க.
- பட்டியலிலிருந்து பார்க்க விரும்பும் **Slide Show** –ஐ தேர்வு செய்க.
- **Start** பொத்தானை கிளிக் செய்க.

குறிப்பு : தனிப்பயன் சில்லுக்காட்சியைத் தேர்வு செய்து தொடங்குவதற்காக Start பொத்தானை கிளிக் செய்யும்போது **Use Custom Slide Show** பெட்டி தேர்வு செய்யப்பட்டுள்ளதா என உறுதிப்படுத்திக் கொள்ள வேண்டும். (படம் 9.35)

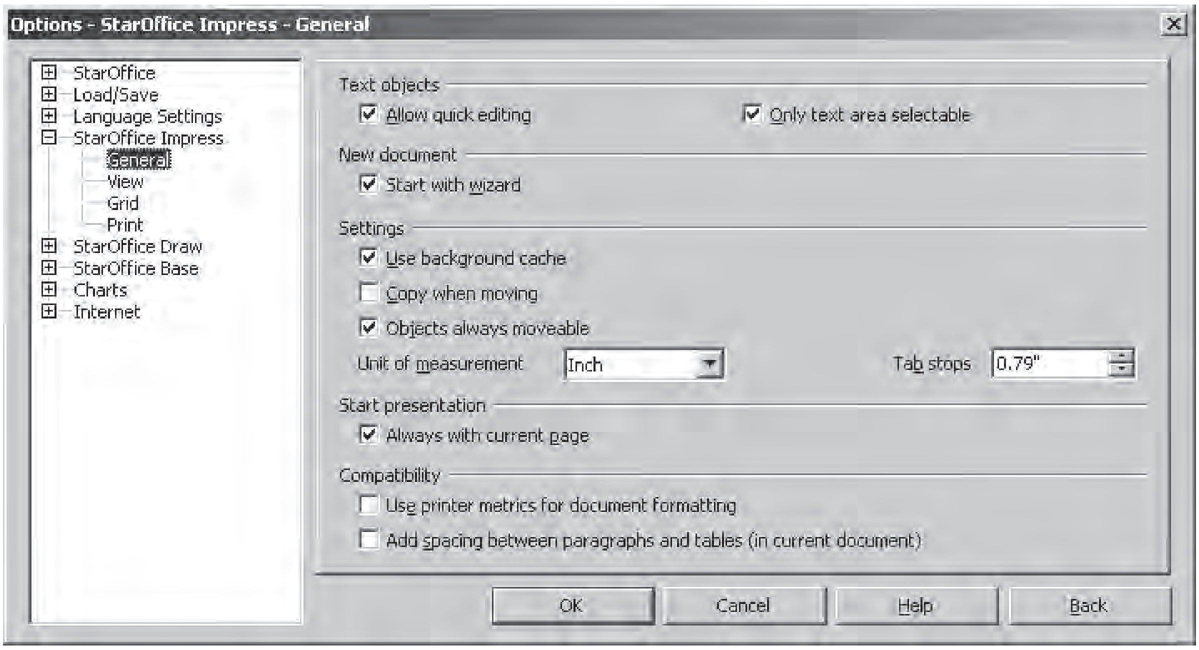


படம் 9.35 Custom Slide show

9.4.1 சில்லுக்காட்சியை ஓட்டும் வகைகள்

தற்போது திரையில் உள்ள சில்லுவிலிருந்து சில்லுக்காட்சியைத் தொடங்க

1. Tools → Options → Star Office Impress → General இணைப்புகளைத் தேர்வு செய்க. படம் 9. 36–இல் உள்ள சன்னல் திரை தோன்றும்.
2. Start Presentation பகுதியில் Always With Current Page பெட்டியைத் தேர்வு செய்க.

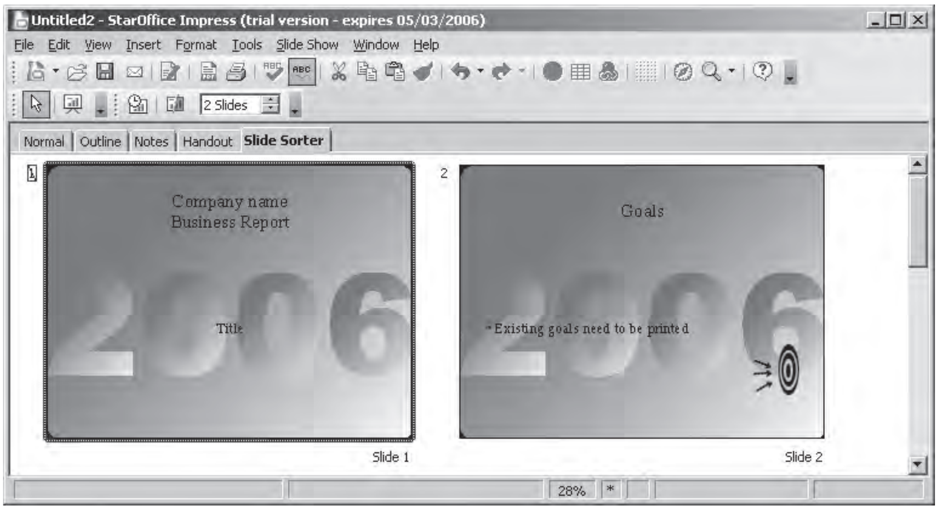


படம் 9.36 General Settings in StarOffice Impress

9.4.2 சில்லுவை மறைக்கும் வழிகள்

- சில்லுக் காட்சியிலிருந்து மறைக்க விரும்பும் சில்லுவைத் தேர்வு செய்க.
- Slide Show→Show / Hide Show இணைப்புகளைத் தேர்வு செய்

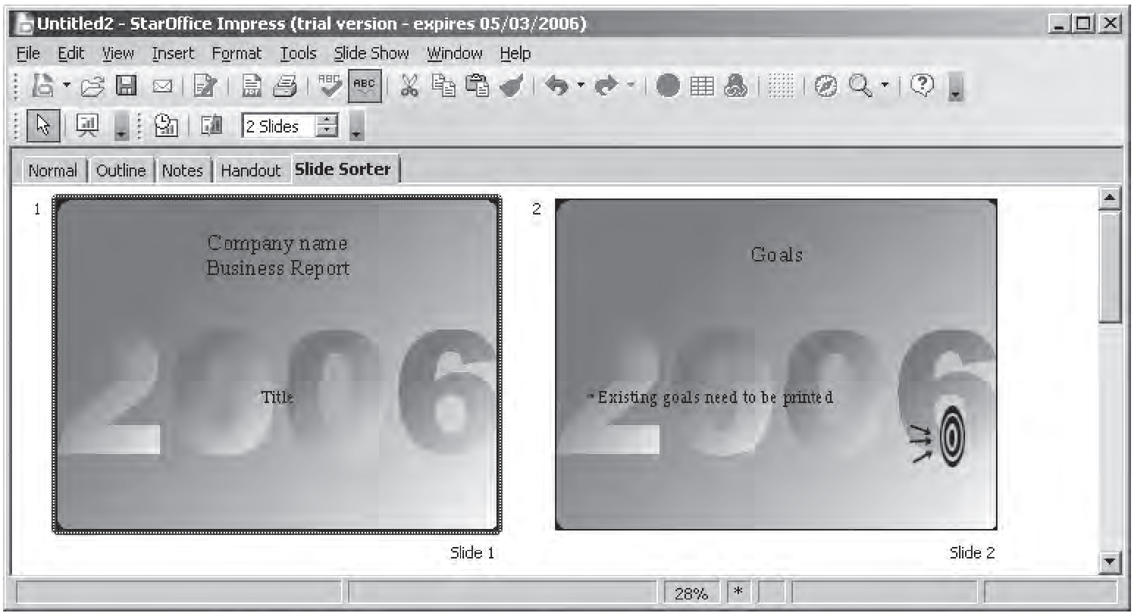
படம் **9.37**–இல் காட்டப்படுள்ளதைப் போல் தேர்வு செய்து சில்லு எண் நீக்கப்படும். ஆனால் அந்த சில்லு நிகழ்ச்சித் தொகுப்பிலிருந்து நீக்கப்படவில்லை என்பதைக் கருத்தில் கொள்க.



படம் 9.37 Hidden Slide in Slide View

9.4.3 மறைத்த சில்லுவைத் திரும்பக் காட்டுதல்

Slide Pane–லிருந்து திரும்பக் காட்ட விரும்பும் மறைத்த சில்லுவை தேர்வு செய்க. Slides Show→ Show / Hide Silde இணைப்புகளைத் தேர்வு செய்தால் மறைத்த சில்லு படம் **9.38** –இல் காட்டியுள்ளது போல் தோன்றும்.



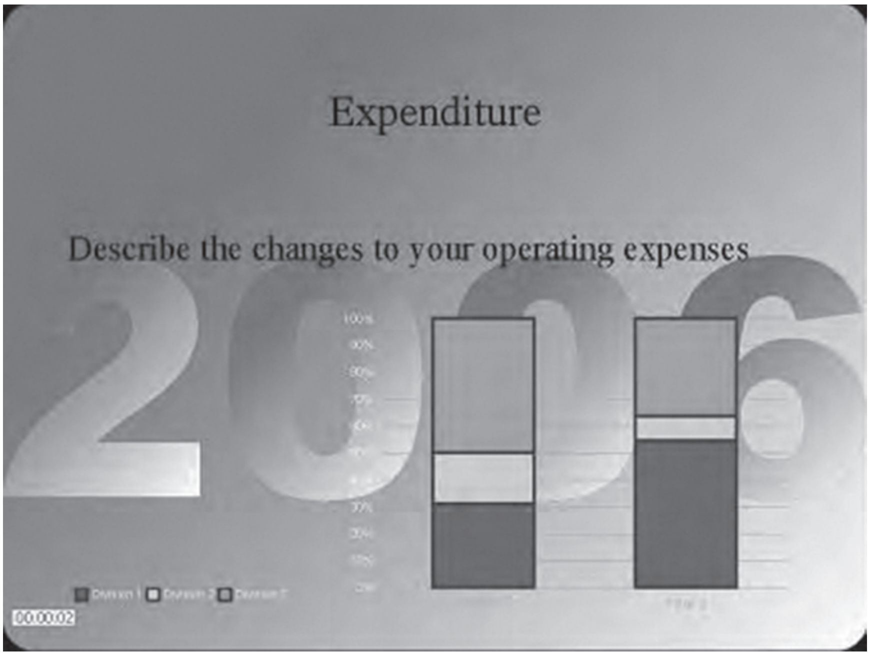
படம் 9.38 Shown Slide in Slide view

சில்லு மாற்றங்களின் ஒத்திகை நேரங்கள்

சில்லுகளின் ஓட்ட நேரத்தை Star Office பதிவு செய்கின்றது. எனவே, நம்மால் தானியங்கு சில்லு மாற்றங்களின் ஓட்ட நேரத்தை வரையறுக்க முடிகிறது.

சில்லுவின் காட்சி நேரத்தை பதிவு செய்ய

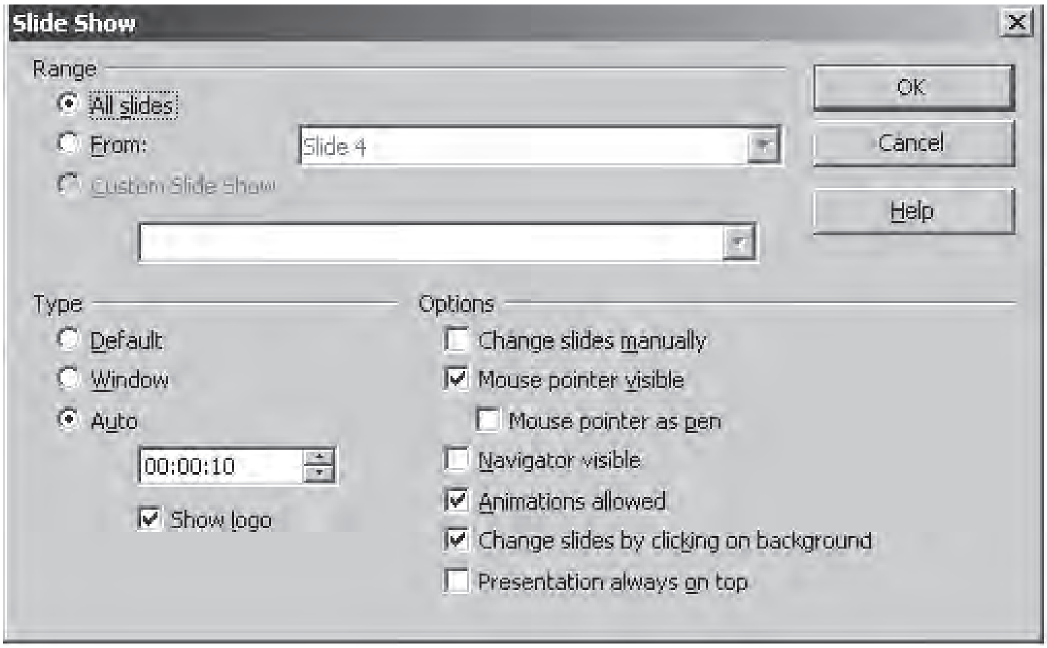
- ஒரு நிகழ்த்தலைத் திறந்து Normal பார்வைக்குச் செல்லவும்.
- Slide Show பட்டியிலிருந்து Rehearse Timings இணைப்பின் மூலம் காட்சியைத் தொடங்கு. படம் 9.39–இல் காட்டியுள்ளவாறு சில்லுவின் காட்சியுடன், திரையின் அடியில் இடக்கோடியில் ஒரு கடிகாரத்தைப் பார்க்கலாம்.
- காட்சி அடுத்த சில்லுவிற்கு மாறும் நேரத்தில் கடிகாரத்தைக் கிளிக் செய்து, நேரத்தை அறிக. நிகழ்த்தலின் அனைத்து சில்லுகளுக்கும் இதைச் செய்க.
- Star Office ஒவ்வொரு சில்லுவின் ஓட்ட நேரத்தையும் பதிவு செய்திருப்பதைக் காணலாம்.
- முழு நிகழ்த்தலை தானே திருப்பி ஓடச் செய்ய வேண்டுமெனில், Slide Show→Slide Show Settings பட்டிகளைத் திறக்கவும். படம் 9.40–இல் உள்ளதைப் போல சன்னல்திரை தோன்றுகிறது.
- Auto–ஐ தேர்வு செய்து ok பொத்தானை கிளிக் செய்க.



படம் 9.39 Rehearse Timings in Slide Show

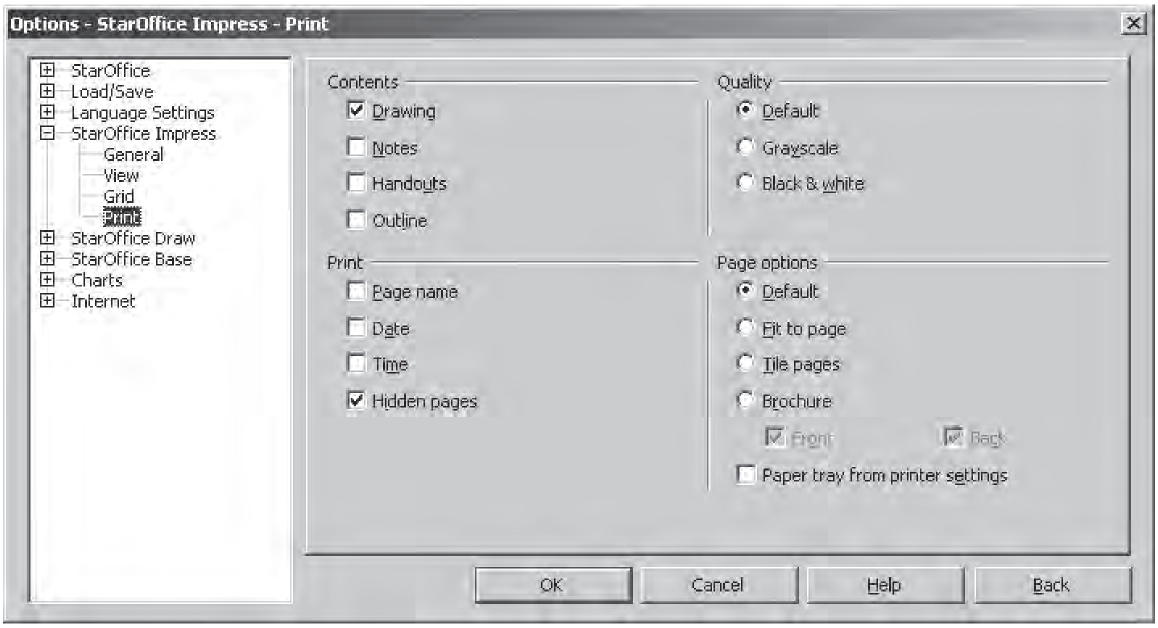
9.5 நிகழ்த்தலை அச்சிடல்

Star Office Impress–க்கு கொடாநிலை அச்சிடும் வாய்ப்பை (default option) வழங்க Tools →Options→ Star Office Impress→ Print ஆகிய இணைப்புகளைத் தேர்வு செய்க. படம் 9.41–இல் காட்டியுள்ளது போன்ற சன்னல் திரை தோன்றும். இந்த சன்னல் திரையில் செய்யும் மாற்றங்கள் அனைத்தும் அச்சுப்பொறிக்கு கொடாநிலை அமைவாகக் கொள்ளப்படும்.



படம் 9.40 Slide Show Type

நிகழும் நிகழ்த்தலுக்கு அச்சுப்பொறி அமைவுகளை நிறுவ File → Print என்னும் இணைப்புகளைத் தேர்வு செய்க. படம் 9.42–இல் காட்டியுள்ள திரை தோன்றும். Option இணைப்பை கிளிக் செய்து அச்சுப் பொறியின் அமைவுகளை முடிவு செய்க. அமைவுகளின் இந்தத் தேர்வு கொடாநிலை தேர்வுக்கு மேல் நின்று செயல்படும்.

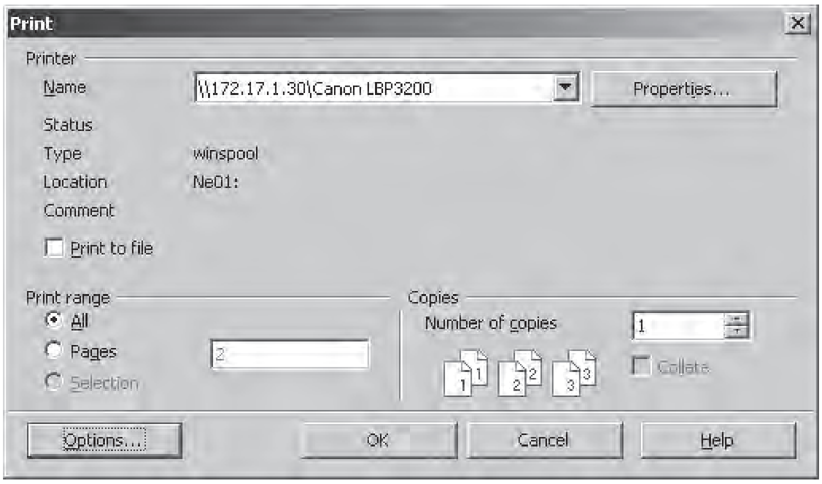


படம் 9.41 Printer Options window

ஒரு வரிசை சில்லுவை அச்சிடல்

File→Print இணைப்புகளைத் தேர்வு செய். Print range பகுதியில் Pages பொத்தானை கிளிக் செய். Pages பெட்டியில் அச்சிடப்பட வேண்டிய சில்லுகளின் வரிசை எண்களைக் கொடு. பிறகு

OK பொத்தானை கிளிக் செய். எடுத்துக்காட்டாக **3–**வது, **5–**வது சில்லுகளை அச்சிட, **3,4,5** அல்லது **3–5** என்று படம் **9.43 –**இல் காட்டியுள்ளவாறு Pages பெட்டியில் கொடு.

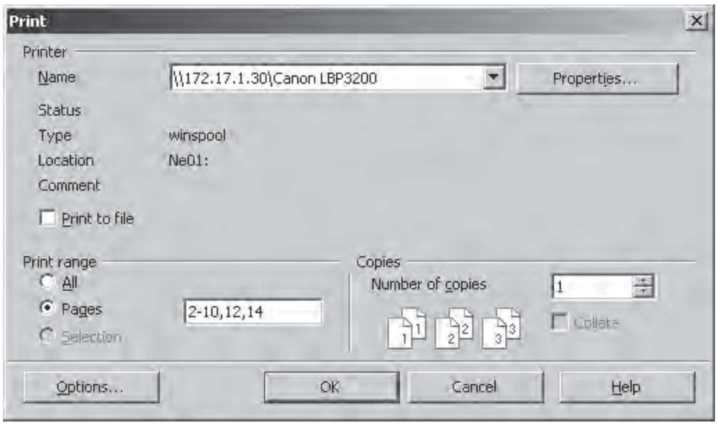


படம் 9.42 Printer Options window for Current Presentation

Star Office-இல் Print File Directly குறும்படத்தை செயற்பட்டை (Function Bar)யிலிருந்து கிளிக் செய்து அச்சிடலாம். உரையாடல் பெட்டித் தோன்றாமலேயே, அச்சிட வேண்டிய ஆவணம் உடனே அச்சிடப்படும். இதற்கு மாறாக, File→Print இணைப்புகளை அல்லது Ctrl + P பொத்தான்களைத் தேர்வு செய்தால், அச்சிடும் உரையாடல் பெட்டி தோன்றும். இதில் உள்ள அச்சிடும் வாய்ப்புகளைத் தேர்வு செய்து அச்சிடலாம். எடுத்துக்காட்டாக , ஒரு பக்கம் மட்டுமே, அல்லது ஒரு வரியில் உள்ள பக்கங்கள் அல்லது எல்லா பக்கங்கள் எனத் தேர்வு செய்து அச்சிட்டுக் கொள்ளலாம்.

ஒரு பக்கத்தில் நிறையுமாறு ஒரு சில்லுவை அச்சிடல்

சில்லுவின் அளவை ஒரு பக்கத்திற்குள் அடங்குமாறு குறைத்து அல்லது பெரிதுபடுத்தி அச்சிட்டுக் கொள்ளலாம். இதைச் செய்ய, அச்சிட விரும்பும் பக்கத்தைத் திறந்து கொள்க. இயல்பான காட்சியில், Format→ Page இணைப்புகளையும், Page tab இணைப்பைப் பிறகும் தேர்வு செய் (படம் **9.44**) பிறகு Layout Settings பகுதியில் Fit object to paper format பெட்டியைத் தேர்வு செய். paper format பகுதியில் ஒரு படிவத்தைத் தேர்வு செய்து OK பொத்தானை கிளிக் செய். இப்பொழுது சில்லு ஒரு பக்கத்திற்குள் அடங்குமாறு அளவு செய்யப்பட்டு அச்சிடப்படும்.

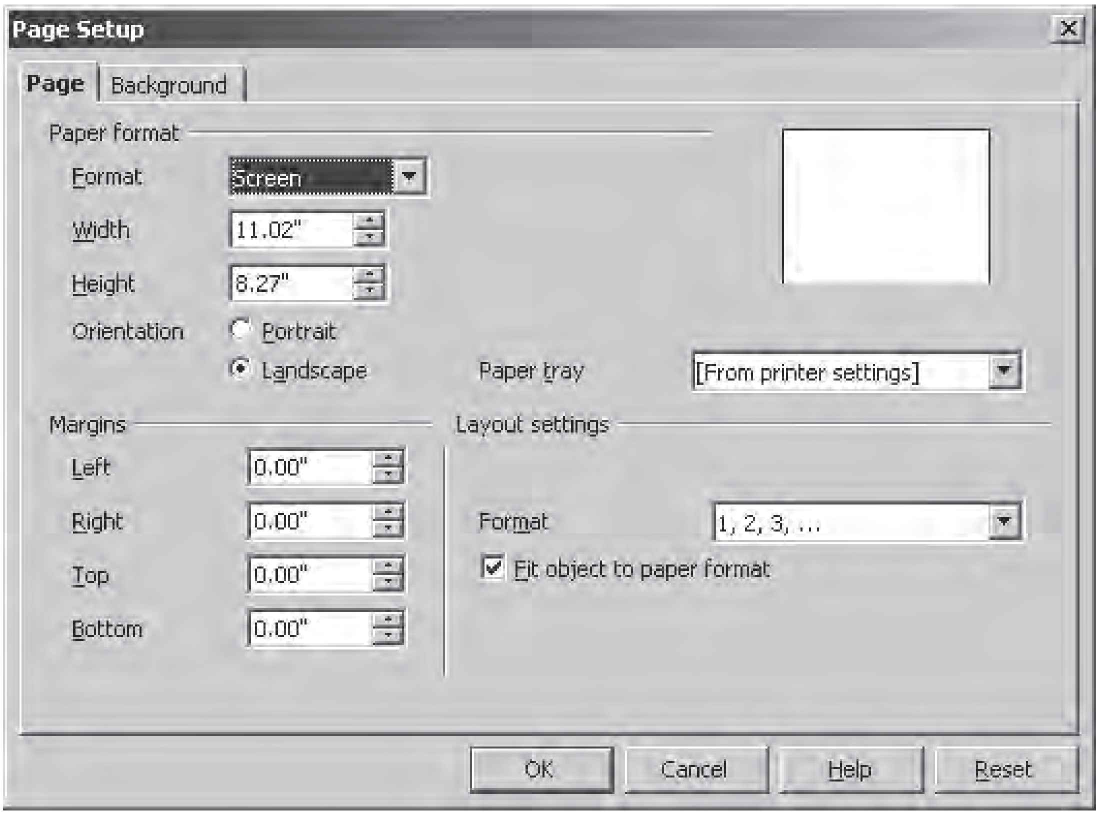


படம் 9.43 Printing Options for printing a range of slides

கை அறிக்கைகளை (Handouts) உருவாக்கி அச்சிடல்

கை அறிக்கைகளை வழங்க விரும்பினால், Handout View முறையைப் பயன்படுத்தலாம். கை அறிக்கையில் சில்லுவுடன் குறிப்புகள் கொடுக்கவும் இடம் உள்ளதால், அவை மிகவும் பயனுள்ள நிகழ்த்தலாக இருக்கும்.

View bar–லிருந்து Handout View –ஐத் திறக்கவும். பிறகு பணிப்பட்டையிலிருந்து Layout Pane–ஐத் திறக்கவும். படம் 9.45 –இல் காட்டியுள்ளவாறு, ஒரு பக்கத்தில் எத்தனை சில்லுகளை அச்சிட வேண்டும் என்பதைத் தேர்வு செய்யவும். பிறகு அவற்றை அச்சிட File→Print இணைப்புகளைத் தேர்வு செய்து, அச்சு அமைவுகளை வரையறுக்க Options பொத்தானை கிளிக் செய்க.



படம் 9.44 **Page Setup for Printing a Slide to fit paper size**

அச்சிடும் குறிப்புகளை ஏற்படுத்துதல்

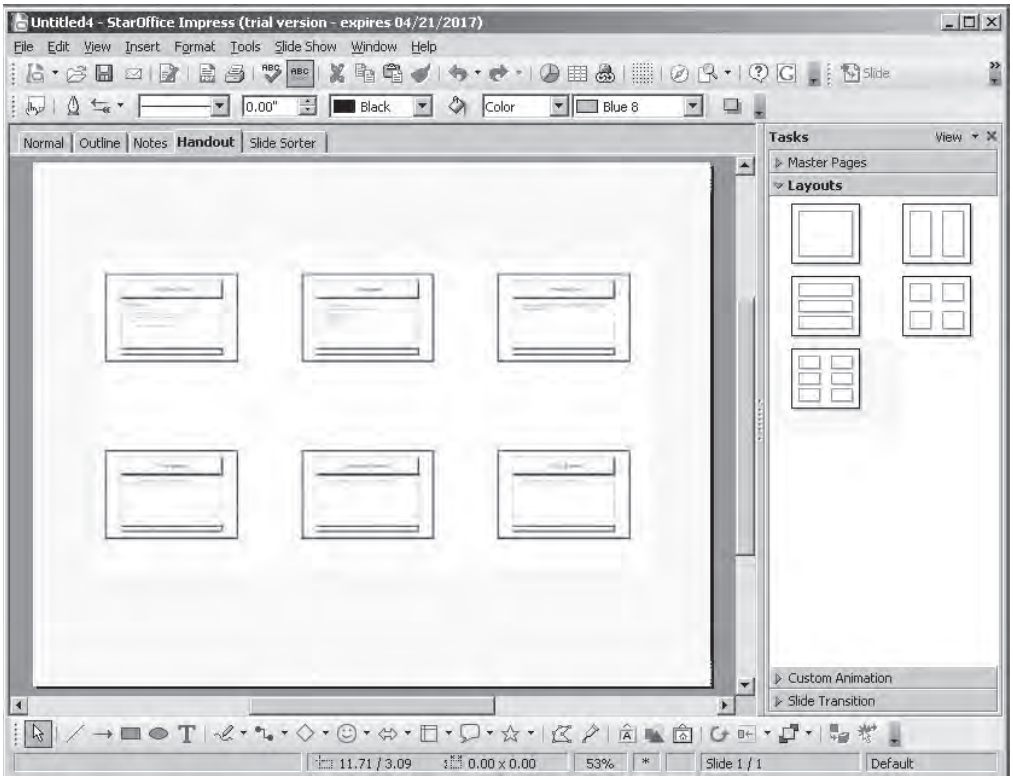
Notes View பகுதியில் அச்சிட வேண்டிய குறிப்புகளை ஒவ்வொரு சில்லுவிலும் கொடுக்கவும். கொடுத்த குறிப்புகள் படம் 9.46–இல் உள்ளதைப் போலத் தோன்றும். View பட்டியிலிருந்து Notes View பொத்தானைப் பயன்படுத்தி குறிப்புகளைப் பார்க்கலாம்.

சில்லுக்குறிப்புகளை அச்சிடல்

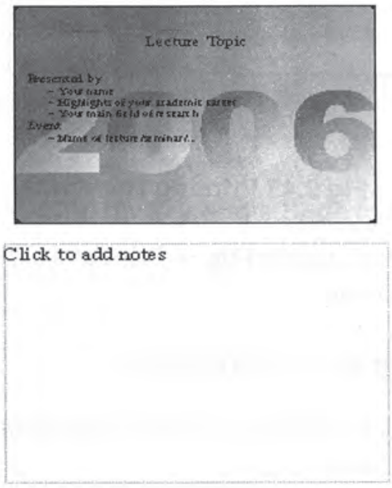
1. File → Print இணைப்புகளைத் தேர்வு செய்து Options–ஐ கிளிக் செய்க.
2. Print Options உரையாடல் பெட்டியில் Contents பகுதியிலிருந்து Notes இணைப்பைத் தேர்வு செய்து OK பொத்தானை கிளிக் செய்க.

3. Print உரையாடல் பெட்டியிலிருந்து அச்சிட வேண்டிய சில்லுகளைத் தேர்வு செய்து OK பொத்தானை கிளிக் செய்க.

அச்சிடும் பக்கத்தில் குறிப்புகள் அச்சிடப்பட வேண்டியதில்லை எனில்,Print Options உரையாடல் பெட்டியில் Notes இணைப்பை நீக்க வேண்டும் என்பதை நினைவில் கொள்க. Print Option–இல் கொடுக்கும் அமைவுகள் Settings, தேர்வு செய்யப்பட்ட அந்த பக்கங்களுக்கு மட்டுமே உரியவை. குறிப்பிட்ட சில அமைவுகள் நிகழ்த்தலில் உள்ள எல்லா சில்லுகளுக்கும் தேவையெனில் Tools→Options→StarOffice Impress→Print இணைப்புகளைத் தேர்வு செய்க.



படம் 9.45 Printing Handouts



படம் 9.46 Entering Notes for Printing

உலவுதல் (Navigation)

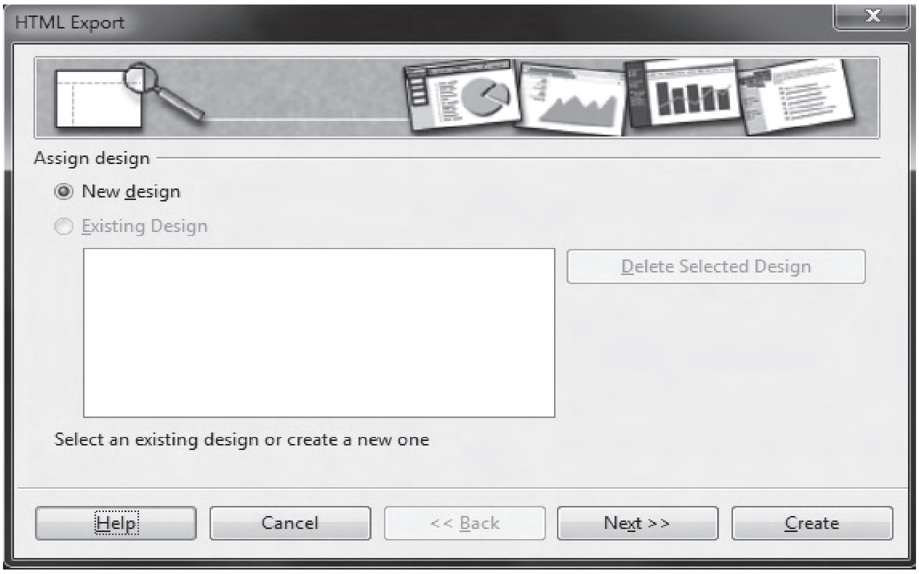
Edit→Navigator இணைப்புகளைத் தேர்வு செய்து அல்லது செயற்பட்டியில் (Function bar) உள்ள உலவி குறும்படத்தை கிளிக் செய்து, ஒரு சில்லுவிலிருந்து இன்னொரு சில்லுக்கு விரைவில் சென்று பார்க்க இயலும். கிளிக் செய்தவுடன், படம் **9.47**–இல் காட்டியுள்ள சன்னல்திரை தோன்றும். அதில் எந்த சில்லுவைப் பார்க்க வேண்டுமோ அதன் மீது இருமுறை கிளிக் செய்தால் அந்தச் சில்லுக்குத் தாவிச் செல்லும்.

படம் 9.47 Navigator window

நிகழ்த்தலை HTML வடிவாக்கல்

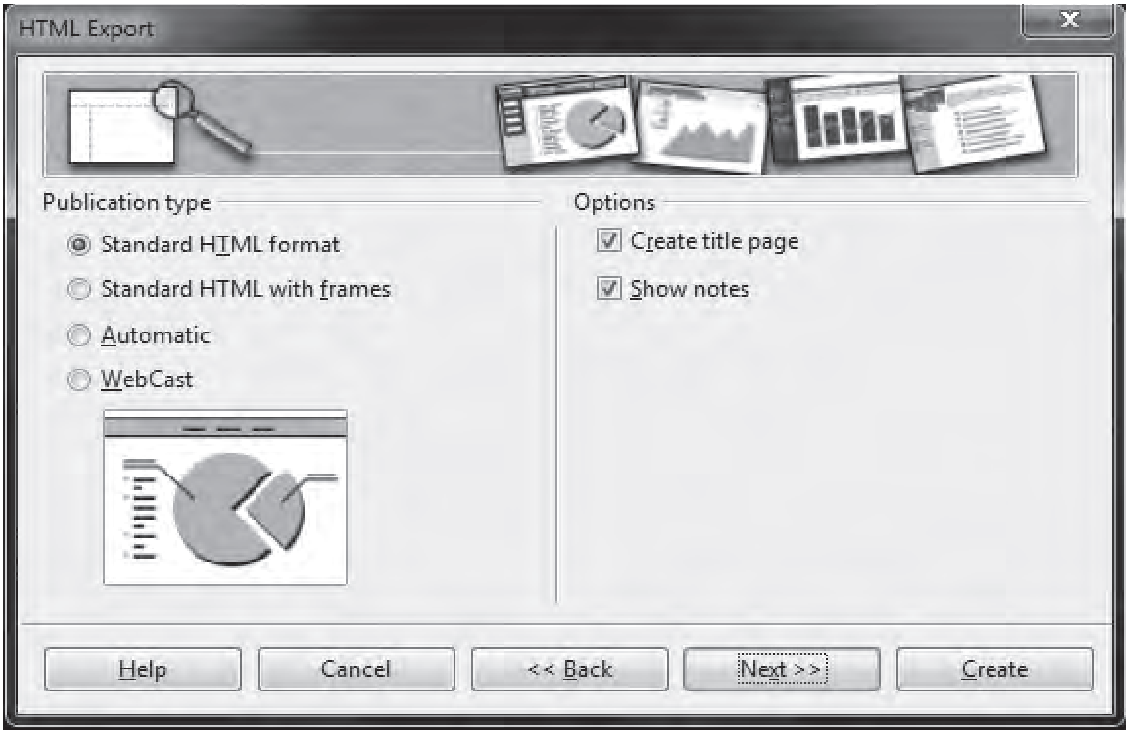
Star Office Impress HTML நிகழ்த்தல் பக்கங்களை வடிவமைக்க உதவுகிறது. பல HTML பக்கங்களை ஒன்றுக்கொன்று இணைக்கலாம். அவற்றில் வரைபடங்கள் GIF அல்லது JPEG நிழற்படங்களாக சேமிக்கப்படுகின்றன. Star Office –இல் உள்ள உரைக்கோப்பு மூலம் HTML பக்கங்களில் தலைப்பு கொடுத்தல், போன்றவற்றைச் செய்யலாம். இதனைச் செய்ய,

- File →Export இணைப்புகளைத் தேர்வு செய்க.
- HTML ஆவணமாக கோப்பின் வடிவமைப்பைத் தேர்வு செய்க.
- கோப்பிற்கு பெயர் கொடுத்து save பொத்தானை கிளிக் செய்து சேமிக்கவும்.
- HTML Export பக்கம் திறக்கிறது. (படம் **9.48**).



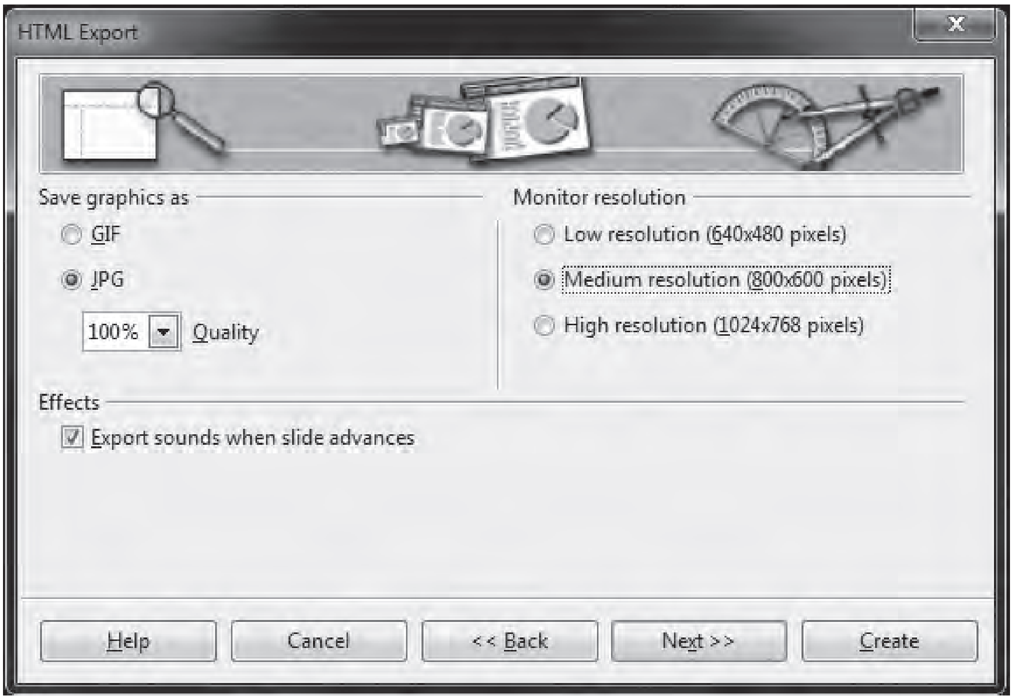
படம் 9.48 HTML Export Wizard

- New Design இணைப்பைத் தேர்வு செய்து Next பொத்தானை கிளிக் செய். படம் **9.49–**இல் உள்ள திரை தோன்றும்.
- Publication Type Option–இல் Standard HTML format இணைப்பை தேர்வு செய்து சட்டங்கள் இன்றி HTML பக்கங்களை உருவைக்கவும்.
- Options பெட்டியைத் திறந்து தலைப்புப் பக்கத்தை அமைத்துக் கொள்ளலாம்.



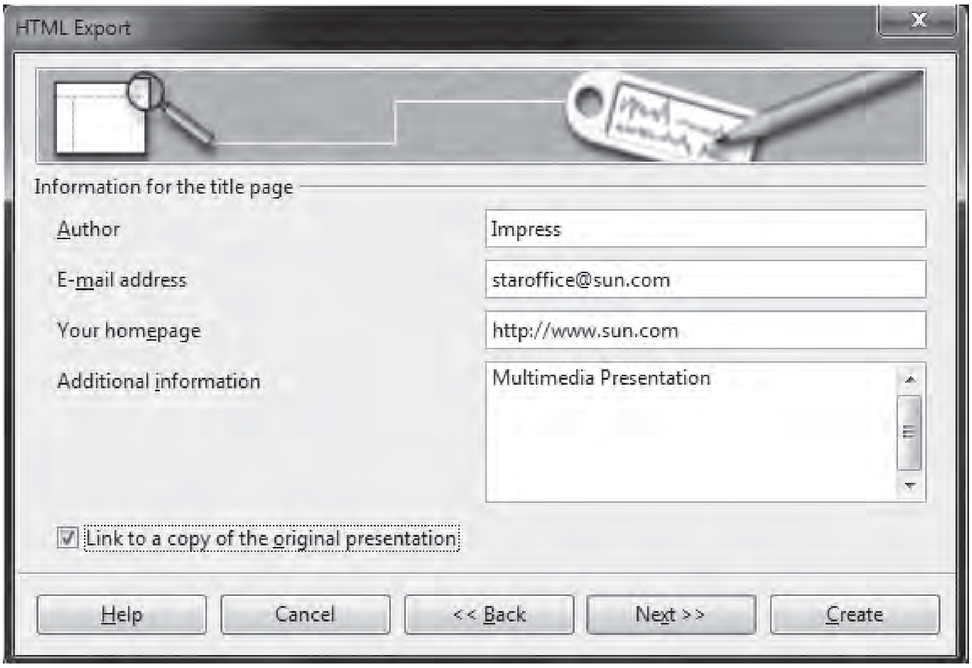
படம் 9.49 Choosing Publication Type

படம் 9.50–இல் காட்டியுள்ளவாறு GIF அல்லது JPEG படிவங்களைத் தேர்வு செய்து படங்கள்அல்லது வரைபடங்களை HTML நிகழ்த்தலில் சேமித்துக் கொள்ளலாம். பிறகு Next பொத்தானை கிளிக் செய்வது இயல்புத் தரமான Medium Resolution –ஆக அமைக்கவும்.



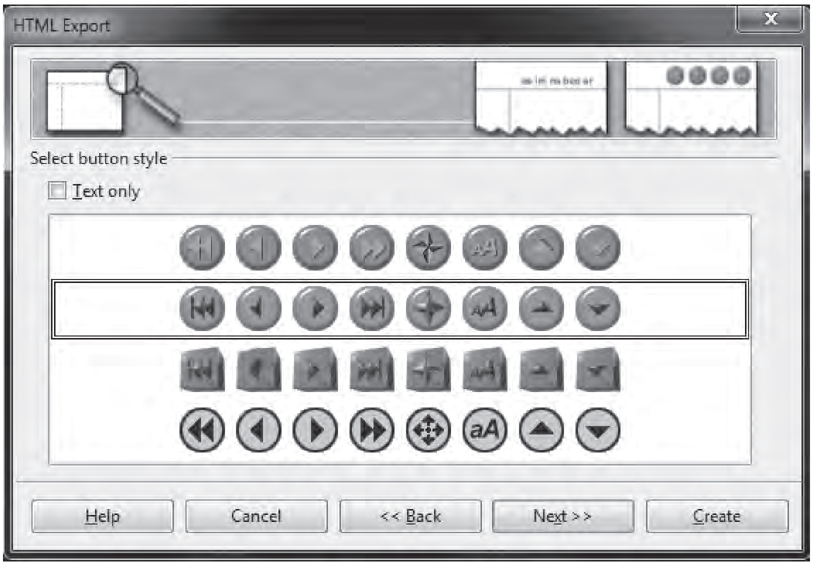
படம் 9.50 Setting Display Size

அடுத்து Next பொத்தானை கிளிக் செய். படம் 9.51 –இல் உள்ள திரை தோன்றும். அதில் விரும்பினால் தொடர்புச் செய்திகளைக் கொடுக்கலாம்.



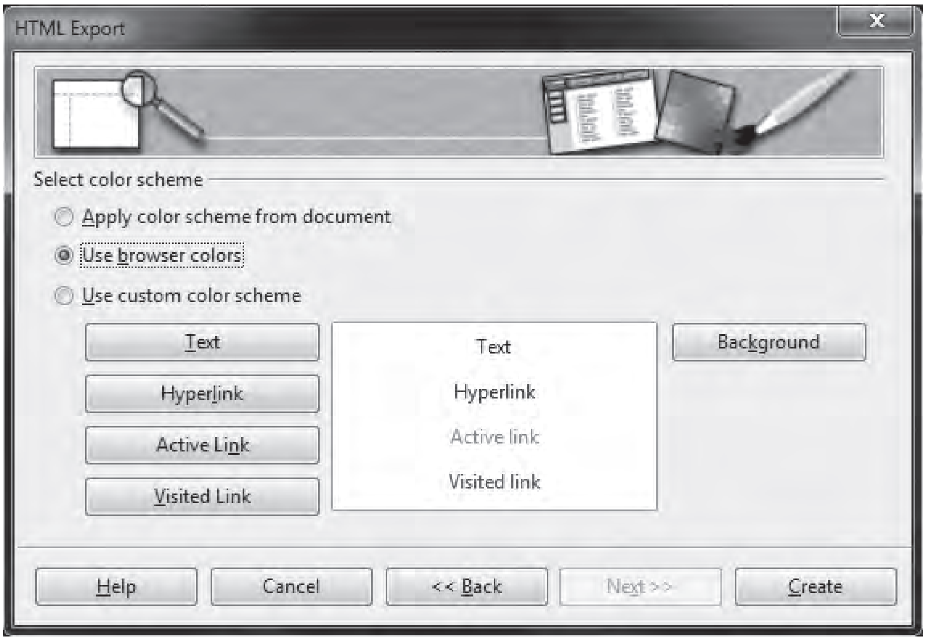
படம் 9.51 Entering Contact Information

இந்தத் திரை உலவிப்பட்டியில் (Navigation bar) இட வேண்டிய வரைபடப் பொத்தான்களை (குறும்படங்கள்) எடுத்து இட உதவுகிறது. நமது நிகழ்த்தலின் பின்னணிக்கு ஏற்ற பொத்தான்களைத் தேர்வு செய்து, Next பொத்தானை கிளிக் செய்தால் உலவிப் பட்டியில் குறும்படங்கள் இடப்பட்டுவிடும். (படம் 9.52).



படம் 9.52 Selecting Button Style

HTML பக்கங்களுக்கு வண்ணமிட, Use Browser Colors இணைப்பைத் தேர்வு செய்து Create பொத்தானைக் கிளிக் செய்க. (படம் 9.53) நமது தேர்வுகளையெல்லாம் HTML வடிவாகச் சேமிக்க வேண்டிய அறிவிப்புத் தோன்றும். வடிவமைக்குப் பெயரிட்டு Save பொத்தானை கிளிக் செய்யவும். குறிப்பிட்ட தகவல் தொகுதியில் (Directory) HTML பக்கங்கள் உருவாக்கப்பட்டிருக்கும். பிறகு முதல் பக்கத்தைத் திறந்து பிறபக்கங்களுக்கு உலவலாம்.



படம் 9.53 Selecting Color Scheme

சுருக்கம்

- சில்லுக்காட்சி என்பது ஒரு குறிப்பிட்ட பொருள் பற்றிய செய்திகளை வரைபடச் சில்லுகளாகவோ, ஆவணப் பக்கங்களாகவோ கொடுக்கும் ஒரு காட்சியாகும்.
- ஒரு பொருள் பற்றி அரங்கில் உரை நிகழ்த்த நிகழ்த்தல் பயன்படுகின்றது.
- சில நிகழ்த்தல்கள், பேச்சாளர் இன்றி, படங்களைக் கொண்ட சில்லுக்காட்சிகள் மூலம் மட்டுமே நிகழ்த்தப்படலாம்.
- நிகழ்த்தலை அடிப்படையிலிருந்து உருவாக்கலாம்; முன்பே செய்யப்பட்டுள்ள முன் வரைவுத் தொகுப்புகளிலிருந்தும் உருவாக்கலாம்.
- தானியங்கும் நிகழ்த்தல் சில்லுகளை ஒரு குறிப்பிட்ட வேகத்தில் Escape சாவியை அழுத்தும் வரை தானே வழங்கிக் கொண்டிருக்கும்.
- Normal, Outline, Notes, Handout, Slide Sorter, Views போன்ற வசதிகள், சில்லுகளைப் பயன்படுத்தி பல தேவைகளுக்கு எளிதாக செயல்பட உதவுகின்றன.
- நிகழ்த்தல், பல வகையான படங்கள், ஒளிப்படக் காட்சிகள், ஒலி, வரைபடங்கள், அட்டவணைத்தாள்கள் மற்றும் பிற OLE பொருள்கள் ஆகியவற்றைக் கொண்டுள்ளன.
- நிகழ்த்தலை இணையப் பக்கங்கள், PDF, SWG, JPEG போன்ற பல வடிவமைப்புகளுக்குள் எடுத்துச் செல்லலாம்.
- Star Office Impress–இல் சில்லுகளைத் தேவைக்கேற்ப, வண்னங்கள், சாய்வமைப்புகள், வரிவேய்தல், Bitmap நிழற்பட வகைகள் போன்ற வசதிகளைப் பயன்படுத்தி வடிவமைக்கலாம்.
- முதன்மைச் சில்லு, ஒவ்வொரு சில்லுகளின், உரை வழங்கும் பாணி, வரைச்சட்டம் (Outline), பின்னணி ஆகியவற்றை முடிவு செய்கின்றது.
- Custom Animation Effects–ஐப் பயன்படுத்தி சில்லுக்குள் உரைப் பொருள்கள் இடமிருந்து நகர்தல், மெதுவாகத் திரையில் தோன்றுதல் போன்ற விளைவுகளை வடிவமைக்கலாம்.
- Custom Slide Show–ஐப் பயனபடுத்தி நிகழ்த்தலின் போது முக்கியமான சில்லுகளை மட்டும் காண்பிக்கலாம்.
- Star Office Impress சில்லுக் காட்சியின் பொழுது சில்லுகளை தொடக்கத்திலிருந்து பார்க்கவும், சில்லுகளை மறைக்கவும் வழி வகுக்கின்றது.
- நிகழ்த்தலில் உள்ள பல சில்லுகளை ஒரு பக்கத்தில் அல்லது பல பக்கங்களில் கை அறிக்கைகளாக அச்சிட்டு வழங்கலாம்.
- Star Office Impress, Media Player மூலம் ஒளிப்படக் காட்சி, ஒலிக் கோப்புகள் ஆகியவற்றை முன் பார்வையாகப் பார்க்கவும், நிகழ்த்தலில் சில்லுகளாகச் சேர்க்கவும் உதவுகிறது.
- Star Office Impress–இல் இடப்பட்டுள்ளMedia Player பல வகையான ஒலி, ஒளிக்காட்சிக் கோப்பு வடிவமைப்புகளை வழங்குகிறது.

பயிற்சி

I. கோடிட்ட இடங்களை நிரப்புக.

1. சில்லுக்காட்சி என்பது சில்லுகளைக் கொண்டு நிகழ்த்தல் ஒரு _______________.
2. நிகழ்த்தலை அடிப்படையிலிருந்தோ அல்லது _______________ இருந்தோ உருவாக்கலாம்.
3. ஐந்து வகையான காட்சிக் கூறுகளாவன: Normal, Outline, Notes _______________ மற்றும் ___________________.
4. ___________________ சில்லுவில் பொருள்களுக்கு விளைவுகளை வழங்குகிறது.
5. ___________________ View Model –இல் தலைப்புகள் மற்றும் துணைத் தலைப்புகளுடன் சில்லுவின் எல்லா தலைப்புகளும் ஒரு பட்டியலில் தோன்றும்.
6. சாவிப் பலகையில் உள்ள___________________ சாவி அழுத்தி சில்லுவின் அளவைப் பெரிதுபடுத்தலாம்.
7. இன்னொரு பொருளுக்குள் உள்ள ஒரு பொருளைத் தேர்வு செய்ய ___________________ சாவியைப் பிடித்து பொருள் மீது கிளிக் செய்ய வேண்டும்.
8. ________________ Slide தலைப்பு (title) மற்றும் வரைச்சட்டம் (outline) ஆகியவற்றின் வடிவமைப்பு முடிவு செய்கிறது.
9. ________________ சன்னல் திரை விரைவாக ஒரு சில்லுவிலிருந்து இன்னொரு சில்லுவிற்கும், திறந்துள்ள கோப்புகளுக்கிடையே நகரவும் உதவுகிறது.
10. ஒரு சில்லுவில் ஒலி, ஒளிக்காட்சி கோப்புகளை முன்னதாகப் பார்க்க _____________ இணைப்பு உதவுகிறது.

II. கீழ்க்காணும் வினாக்களுக்கு விடை தருக.

1. Impress – ஐ வரையறுத்து, நிகழ்த்தலை உருவாக்குவது பற்றி விவரி.
2. Impress – இன் செயற்கூறுகளை பட்டியலிட்டு விவரி.
3. சில்லுக் காட்சியில் விளைவுகளை உண்டாக்குவதற்கான செய்முறைப் படிகளை விவரி.
4. நிகழ்த்தலை இணையப் பக்கங்களாக வழங்கச் செய்ய வேண்டிய செயற்கூறுகளை விவரி.
5. தனிமயமாக்கப்பட்ட அசைவுப்படம் என்றால் என்ன ? சில்லுகளில் அசைவுப் படக்காட்சிகளை சேர்ப்பது பற்றிய செயல்பாட்டை விளக்குக.
6. சில்லு மாற்றல் என்றால் என்ன ? சில்லுகளுக்கு மாற்றல் முறையை வழங்கும் செயல்பாட்டை விளக்கு.
7. நிகழ்த்தலில் படங்கள், ஒளிப்படக்காட்சிகள் , மற்றும் பிற OLE பொருள்களை சேர்ப்பது எப்படி ?
8. ‘Rehearse Timing’ என்றால் என்ன ? விரிவாக விளக்கு.
9. ‘Custom Slide Show’ பற்றி விவரி.

10. கொடுக்கப்பட்ட ஒரு வரிசைச் சில்லுகளை அச்சிடுவது எப்படி என்றும் ஒரு சில்லுவை ஒரு பக்க அளவிற்குள் அச்சிடுவது எப்படி என்றும் விளக்கு.

இணைப்பு

மேற்படிப்பிற்கு

Star Office

Docs.sun.com என்னும் இணையதளத்தில் sun technical document விவரங்கள் இணைய வழிப் பெரும் வகையில் கொடுக்கப்பட்டுள்ளன. Star Office தொடர்பான எல்லா செய்திகளையும் இந்த இணைய தளத்தில் இருந்து பெறலாம்.

அதன் முகவரி: <http://docs.sun.com>.

பாடம் 10

சி++ மொழியின்வழி பொருள்நோக்கு நிரலாக்கக் கருத்துருக்கள் (Object Oriented Concepts using C++)

10.1 பொருள்நோக்கு நிரலாக்க மொழி

(Object Oriented Programming Language)

பலதரப்பட்ட சிக்கல்களுக்குத் தீர்வுகாணும் ஒரு கருவியாகக் கணிப்பொறி விளங்குகிறது. சிக்கல்களுக்கான தீர்வுகள் கணிப்பொறி நிரல்களாகவும் (Programs) பயன்பாட்டு மென்பொருள்களாகவும் (Application Software) நமக்குக் கிடைக்கின்றன. இத்தகைய நிரல்களும் மென்பொருட்களும் ஏதேனும் ஒரு நிரலாக்க மொழியில் எழுதப்படுகின்றன.

ஒரு கணிப்பொறி நிரலானது, உள்ளீடாகத் தரப்படும் தரவுக் கூறுகளின் மீது செயல்படுகின்றது. இந்த உள்ளிட்டுத் தரவுகள் நமக்குத் தேவையான தகவல் கூறுகளாக மாற்றப்பட்டு, கணிப்பொறி நிரலின் வெளியீடாகப் பெறப்படுகின்றன.

தொடக்க கால நிரலாக்க மொழிகளில் உள்ளீட்டு வெளியீட்டுத் தரவுக் கூறுகள் **மாறிகளாகக்** (variables) கையாளப்பட்டன. உள்ளீட்டுத் தரவுக் கூறுகள் **தரவு இனங்களாக** (Data Types) வகைப்படுத்தப்பட்டன. தரவுக் கூறுகளின் மீது நிகழ்த்தப்படவேண்டிய செயல்பாடுகளை கணிப்பொறிக்கு அறிவுறுத்த **கட்டுப்பாட்டுக் கூற்றுகள்** (Control Statements) பயன்பட்டன.

நிரலாக்க மொழிகள் வேறொரு வகையிலும் பயன்படுகின்றன. ஒரு சிக்கலின் தீர்வுப் பற்றிய நமது கருத்துக்களை ஒருங்கமைக்க அவை உதவுகின்றன. தீர்க்கப்பட வேண்டிய சிக்கல்களும் உருவாக்கப்பட வேண்டிய பயன்பாட்டு மென்பொருள்களும் மிகவும் சிக்கலாகிப் போகும்போது, நிரலாக்க மொழிகளின் இந்த பண்புக்கூறு அதிக முக்கியத்துவம் பெறுகின்றது. இத்தகைய செயல்திறன் கொண்ட நிரலாக்க மொழிகள் பல உருவாக்கப்பட்டுள்ளன. இவை கணிப்பொறிக்கு அறிவுறுத்துவதற்கான எளிய வழிமுறைகளைக் கொண்டுள்ளன.

இருப்பினும், ஒரு சிக்கலுக்கான தீர்வினை, தரவு (data), செயல்பாடுகள் (operations) என இருகூறாகப் பிரித்து நோக்குவது, மனிதர்கள் நடைமுறை வாழ்க்கையின் சிக்கல்களுக்கு தீர்வுகாணும் முறையோடு ஒத்திருக்கவில்லை என்பது உணரப்பட்டது.

சி++ போன்ற பொருள்நோக்கு நிரலாக்க மொழிகள், நடைமுறை வாழ்க்கையின் சிக்கலான பிரச்சினைகளை தீர்க்க, பொதுவாக மனிதர்கள் கையாளும் முறைகளை அடிப்படையாகக் கொண்டு வடிவமைக்கப்பட்டுள்ளன. பொதுவாக மனிதர்கள் நடைமுறைப் பிரச்சினைகளைத் தீர்ப்பதற்கு முன், தீர்வுக்கு தேவைப்படும் பல்வேறு தனித்த பொருள்களை அடையாளம் காண்கின்றனர். அதன்பின் அப்பொருள்களுக்கு இடையேயான உறவுமுறைகளை உணர்ந்து கொள்கின்றனர். உறவுமுறை என்பது ‘இது இதன் ஒரு பகுதி’ அல்லது ‘இது இந்த இனத்தைச் சார்ந்தது’ என்பதைக் குறிப்பதாகும். ஒரு பொருளானது வேறொரு பெரிய பொருளின் ஒரு பகுதி என்பதையு ‘ஒரு பொருளானது இன்னொரு பொருளின் இனத்தைச் சார்ந்தது’ என்பதையும் புரிந்து கொள்கின்ற சாதாரணத் திறம்பாடுகளே நடைமுறை வாழ்க்கையின் சிக்கல்களுக்குத் தீர்வு காண்பதில் மிக இன்றியமையாத இடம்

வகிக்கின்றன. தரவுகளையும் அத்தரவுகளின்மீது நிகழ்த்தப்படும் செயல்பாடுகளையும் ஒருங்கு சேர்த்து, சிக்கல்களுக்குத் தீர்வுகாணும் இந்த வழி முறையைப் **பொருள்நோக்கு நிரலாக்கம்** வழங்கியுள்ளது.

வேறு சொற்களில் கூறுவதெனில், தரவுக் கூறுகளின் தொகுதி, வேறு பிற செயற்கூறுகளை அழைக்காமலே செயல்பாடுகளை நிகழ்த்துவதற்கேற்ற சிறு குழுக்களாகப் பிரிக்கப்படுகிறது. இத்தகைய தரவுக்குழு அதற்கான செயல்பாடுகளுடன் சேர்த்து **இனப்பொருள்** – சுருக்கமாக – பொருள் (object) எனப்படுகிறது செயல்பாடுகள், பொருளின் பண்பியல்பை உணர்த்துகின்றன. நடப்புலகப் பொருளைப் கணிப்பொறி நிரலில் எடுத்தாள்வதற்கு இந்தப் **பொருள்** கருத்துரு உதவுகிறது.

எடுத்துக்காட்டாக, ஒரு கணிப்பியை (calculator) நோக்குங்கள். அதற்கு **நிலை** (State), **பண்பியல்பு** (Behaviour) ஆகியவை உள்ளன. **நிலை** என்பது, நீள அகலம், பொத்தான்கள், காட்சித்திரை, செயற்குறிகள் போன்ற அதன் புறத்தோற்றத்தைக் குறிக்கின்றது. **பண்பியல்பு** என்பது, கூட்டல், கழித்தல், நினைவகத்தில் சேமித்தல், நினைவகத்தை அழித்தல் போன்ற அதன் செயல்பாடுகளை உணர்த்துகின்றது.

படம் 10.1 அடிப்படைக் கணிப்பி

பொருள்நோக்கு நிரலாக்க மொழியில் ஒரு **கணிப்பி** என்பது இவ்வாறு நோக்கப்படுகிறது.

பொருள் – கணிப்பி
தரவுகள் :
Number1,result, operator, Number_backup
செயல்கூறுகள் :
Additon()
Subtraction()
Erase_Memory()
Display_Result()

- ✓ **பொருள்** என்பது தொடர்புடைய செயற்கூறுகள், அச்செயற்கூறுகளுக்கான தரவுகள் ஆகியவற்றைக் கொண்ட ஒரு குழுவாகும்.
- ✓ **பொருள்** என்பது, குறிப்பிட்ட செயற்பரப்புடன் கூடிய, தற்சார்பு கொண்ட ஒருவகைத் துணைநிரல் ஆகும்.

தரவுகளையும் அவை தொடர்பான செயற்கூறுகளையும் **பொருள்கள்** எனப்படும் அலகுகளாகக் குழுவாக்கும் செயலாக்கம் **உறைபொதியாக்கம்** (Encapsulation) என்னும் கருத்துருவுக்கு வழி வகுக்கிறது.

தரவுகளையும் செயற்கூறுகளையும் ஒரு **பொருள் வரையறைக்குள்** ஒன்றாகப் பிணைத்துவைக்கும் செயல்நுட்பம் **உறைபொதியாக்கம்** எனப்படுகிறது.

ஒரு வங்கிக் கணக்கு, மாணவர், பறவை, கார், நாற்காலி போன்றவை நிலை மற்றும் பண்பியல்பு ஆகியவற்றை ஒருங்கே பெற்றுள்ளன என்பதை எளிதாகக் காண முடியும். நடைமுறை வாழ்வில் காணப்படும். உண்மையான பொருள்களோடு உள்ள ஒற்றுமையே கணிப்பொறி நிரலின் பொருள்களுக்கு அவற்றின் சக்தியையும் பயன்பாட்டையும் வழங்குகின்றது. மென்பொருள் நிரல்களில் உண்மைப் பொருள்களை உருவகிக்கும் பணியை பொருள்கள் (objects) எளிமைப்படுத்தியுள்ளன.

எடுத்துக் காட்டுப் பொருள்கள் – வங்கி கணக்கும் மாணவரும்

வங்கிக் கணக்கு	மாணவன்
தரவுகள்:	**தரவுகள்**:
Account number – long int	Date_of_birth – char[10]
Name – char[15]	Name – char[15];
Opening balance –float;	Class, sec char[4];
Account type – char	Marks float
செயற்கூறுகள் :	**செயற்கூறுகள்:**
Accept_Details()	Accept_Details()
Display_Details()	Display_Details()
Update_opening_balance()	Total()
Withdrawls()	Average()
Deposit()	Grading()

10.2 பல்லுருவாக்கம் (Polymorphism)

செவ்வகம், சதுரம், வட்டம், வளைகோடு போன்ற பல்வேறு வடிவங்களை வரையும் பணியை எடுத்துக் கொள்வோம். வெவ்வேறு வடிவங்களை வரைய வெவ்வேறு செயற்கூறுகளை வரையறுக்க வேண்டியிருக்கும். அத்தகைய வரையறைகள் இவ்வாறு இருக்கலாம்.

Draw_Square() Read side Draw required lines	Draw_Rectangle() Read length,breadth Draw required lines	Draw_Circle() Read radius Draw	Draw_Arc() Read Start_angle, End_angle,radius draw

கீழேயுள்ள செயற்கூறுகளை நோக்குக :

Draw(side) – ஒரு சதுரம் வரைவதற்கு

Draw (length, breadth) - ஒரு செவ்வகம் வரைவதற்கு

Draw(radius) - ஒரு வட்டம் வரைவதற்கு

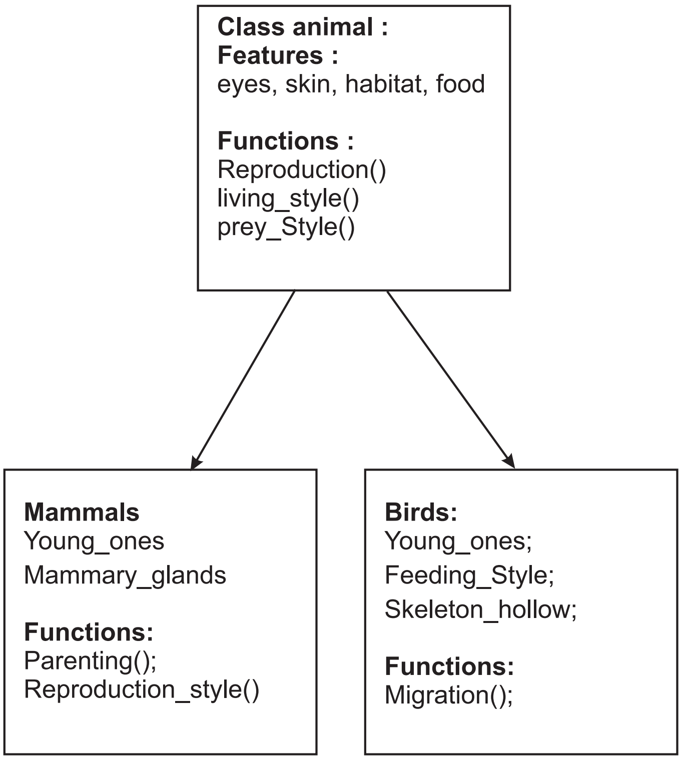
Draw(radius,start_angle,end_angle) – ஒரு வளைகோடு வரைவதற்கு

Draw() என்னும் செயற்கூறு வேறுபட்ட உள்ளீடுகளை ஏற்கிறது. வேறுபட்ட செயல்பாட்டைக் கொண்டுள்ளது. பயனரைப் பொறுத்தவரையில் வேறுபட்ட வடிவங்களை வரைவதற்கு வேறுபட்ட உள்ளீடுகளுடன் Draw () செயற் கூறினைப் பயன்படுத்துவார். Draw() செயற்கூறு வேறுபட்ட உள்ளீடுகளின் அடிப்படையில் வேறுபட்ட முறைகளில் செயல்படும் தன்மையே **பல்லுருவாக்கம்** (Polymorphism) எனப்படுகிறது.

வேறுபட்ட செய்திகளுக்கு மாறுபட்டுச் செயல்படும் ஒரு பொருளின் திறனே **பல்லுருவாக்கம்** என்றழைக்கப்படுகிறது

10.3 மரபுரிமம் (Inheritance)

இனக்குழு (Class) என்னும் தரவினம் நடப்புலகில் நிலவும் ஒரு பொருளை உருவகிக்கிறது. இனக்குழு என்பது பொதுவான பண்பியல்பு கொண்ட உரு பொருள்களின் **வார்ப்புரு** (Template) ஆகும். எடுத்துக்காட்டாக, உயிரினங்களின் குழுவை **விலங்கினம்** என்கிறோம். அதாவது, விலங்குகளை ஒரு இனக்குழுவாக வகைப்படுத்துகிறோம். விலங்குகளைப் பாலூட்டிகள், ஊர்வன, நிலம் நீர் வாழ்வன, பூச்சிகள், பறவைகள் எனப் பலவாறாகப் பிரிக்க முடியும் என்பதை நாமறிவோம். அனைத்து விலங்குகளும் பொதுவான நடத்தையையும் பொதுவான பண்புக்கூறுகளையும் கொண்டுள்ளன. கண்கள், தோல், வாழிடம், உணவு ஆகியவற்றை விலங்குகளின் இயல்புகள் அல்லது பண்புக் கூறுகள் என்கிறோம். இனப்பெருக்கம், வாழும்முறை, இரைகொள்ளும் முறை போன்றவை விலங்குகளின் நடத்தை எனக் கொள்ளலாம். விலங்குகளின் ஒவ்வோர் உட்குழுவும் பொதுவான நடத்தையையும், இயல்பையும் கொண்டிருக்கும். அதே வேளையில் தத்தமக்கே உரிய தனித்த இயல்புகளையும் பாணிகளையும் கொண்டுள்ளன. பாலூட்டிகள், ஊர்வன, நிலம் நீர் வாழ்வன, பூச்சிகள், பறவைகள் ஆகிய உட்குழுக்கள், விலங்கு என்னும் தாய்க்குழுவின் பண்புகளைப் பகிர்ந்து கொள்கின்றன. இதனைப் படமாக இவ்வாறு உருவகிக்கலாம்.



படம் 10.1 மரபுரிமம்

விலங்கு என்பது **அடிப்படை இனக்குழு** (base class) எனப்படுகிறது. பாலூட்டிகள், பறவைகள் என்பவை **தருவிக்கப்பட்ட இனக்குழுக்கள்** (derived classes) எனப்படுகின்றன. தருவிக்கப்பட்ட இனக்குழுக்கள் சக்தி மிக்கவை. காரணம் அவை தமக்குரிய தனித்த இயல்புகளோடு அடிப்படை இனக்குழுவின் செயலாற்றல்களையும் தம்மகத்தே கொண்டுள்ளன. அடிப்படை இனக்குழுவின் பண்புகளை ஈட்டிக் கொள்ளும் தகைமை **மரபுரிமம்** (Inheritance) என்றழைக்கப்படுகிறது.

மரபுரிமம் என்பது தருவிக்கப்பட்ட இனக்குழுவின் செயலாற்றலை மிகுவிக்கிறது. நிரல் குறிமுறையின் மறுபயனாக்கத்தையும் (Code Reusability) சாத்தியமாக்கியுள்ளது.

பொருள்நோக்கு நிரலாக்கத்தின் பலன்கள்

- ✓ நிரல்களில், தரவுகள் மற்றும் செயற்கூறுகளை ஒன்று சேர்த்து பொருள்களாக ஒருங்கமைக்க **இனக்குழு** (Class) என்னும் தரவினம் பயன்படுகிறது.
- ✓ தொடர்பில்லாத வெளிச் செயற்கூறுகள் (இனக்குழுவுக்கு வெளியே வரையறுக்கப்பட்டுள்ள செயற்கூறுகள்) இனக்குழுவுக்குள் இருக்கும் தரவுகளைஅணுக முடியாது. தரவுகளை அணுகத் தடை செய்வதுடன் ஒருபொருளின் மிகத் தேவையான இயல்புகளை மட்டுமே வெளிக்காட்டுகிறது. இத்தகைய **தரவு மறைப்பு** (Data Hiding) அல்லது தரவு **அருவமாக்கம்** (Data Abstraction) தரவுகளுக்குப் பாதுகாப்பை நல்குகிறது.
- ✓ ஒரு செயற்கூறு (Function) அல்லது செயற்குறிக்கு (Operator) பல வரையறைகளைச் சாத்தியமாக்குதல் மூலம் **பல்லுருவாக்கம்** (Polymorphism) மென்பொருள் சிக்கற்பாட்டினை (Complexity) குறைக்கிறது.

✓ **மரபுரிமம்** (Inheritance), ஏற்கெனவே இருக்கும் ஓர் இனக்குழுவின் அடிப்படையில் வேறோர் இனக்குழுவைத் தருவித்துக் கொள்ள வழிவகுக்கிறது. இதன் மூலம் **குறிமுறை மறுபயனாக்கம்** (Code Resuability) சாத்தியமாகிறது. ஏற்கெனவே பயன்பாட்டில் உள்ள மென்பொருளில், புதுப்பிக்கப்பட்டகூறுகளை இணைத்து நாள்தோறும் மாறிக் கொண்டிருக்கும் உலகின் தேவைகளை எதிர்கொள்ள வழி வகுக்கிறது.

10.4 ஒரு நடைமுறை எடுத்துக்காட்டு: வீட்டு நீர்ப்பயன்பாடு

வீட்டுக்கு நீர்வழங்கும் அமைப்பையொத்து ஒரு நிரலை உருவாக்குவதாகக் கொள்வோம். கட்டிடத்திலுள்ள ஒவ்வொரு நீர் வெளியேற்றிகளில் வெளியேறும் நீரின் அளவையும், ஒட்டுமொத்த நீரின் நுகர்வையும் கணக்கிடுவதே இந்த நிரலின் நோக்கம். இந்த நிரலை எழுதுவதற்குத் தேவையான விவரங்கள்: அந்தக் கட்டிடத்தில் நிறுவப்பட்டுள்ள நீர்த் திறப்பிகளின் (Taps) எண்ணிக்கை, ஒவ்வொரு திறப்பியின் வழியாகவும் வெளியேறும் நீரின் அளவு, இறுதியாக மொத்த நீரின் நுகர்வளவு. ஒவ்வொரு திறப்பியையும் ஒரு **பொருளாக** நோக்கலாம். இதனோடு தொடர்புடைய செயற்கூறுகள், நீர்ப்பாய்வை தொடங்குவது நிறுத்துவது மற்றும் குறிப்பிட்ட கால அளவில் நுகரப்படும் நீரின் அளவைத் தருகின்றதாகவும் இன்னும் இவற்றை ஒத்ததாகவும் இருக்கும். இந்தப் பணிகளைச் செய்வதற்குத் திறப்பி என்னும் பொருள், திறப்பி திறந்துள்ளதா, மூடியுள்ளதா என்பதைக் கண்காணிக்கவும், பயன்படுத்தப்பட்ட நீரின் அளவைக் குறிக்கவும், நீர் எங்கிருந்து வருகிறது என்பதை அறியவுமான சான்றுரு மாறிகளைக் (Instance Variables) கொண்டிருக்க வேண்டும். திறப்பி என்னும் பொருளை இவ்வாறு உருவகிக்கலாம்:

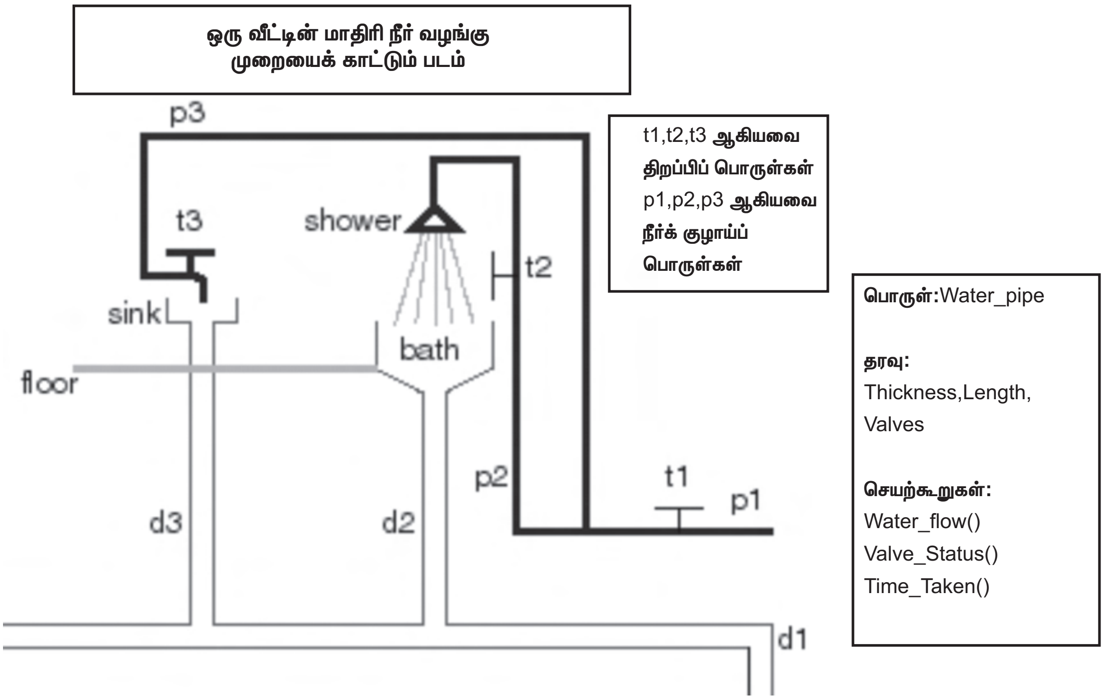
படம் 10.2 வீட்டு நீர்ப்பயன்பாடு

தரவு:
Tap_open,
Qty_water,
Water_source

செயற்கூறுகள்:
Start()
Stop()

படம் 10.3 திறப்பு என்னும் பொருள்

நீரின் பயன்பாட்டை உருவகிக்கும் நிரல், திறப்பிகளுக்கு நீரை வினியோகிக்கும் நீர்க்குழாய் (Waterpipe) என்னும் பொருளையும் கொண்டிருக்கும். அனைத்து நீர்க்குழாய்களையும் திறப்பிகளையும் ஒருங்கிணைக்கும் கட்டடம் (Building) என்னும் பொருளும் இருக்கலாம். எவ்வளவு நீர் நுகரப்பட்டுள்ளது எனக் கட்டடப் பொருளைக் கேட்டோம் எனில், அது ஒவ்வொரு குழாய் மற்றும் அதில் இணைக்கப்பட்டுள்ள திறப்பிகளின் தற்போதைய நிலை என்ன எனக் கேட்டறிந்து சொல்லும். இந்தத் திட்டப்பணியை படம் 10.4–ல் உள்ளவாறு உருவகிக்கலாம்.



படம் 10.4 ஒரு வீட்டின் மாதிரி நீர் வழங்கு முறைமை

நுகரப்பட்ட மொத்த நீரின் அளவு total_amount என்னும் மதிப்பு,

t1. water_consumed()+t2.water_consumed+t3.water_consumed()

என்று கணக்கிடப்படும். தனித்தனி திறப்பியிலிருந்து நுகரப்பட்ட நீரின் அளவு t1.water_consumed(), t2. water_consumed, t3.water_consumed(),...என்று அறியப்படும்.

t1.water_consumed() என்னும் செயற்கூறு அழைப்பு P1 என்னும் குழாய் பொருளுடன் தொடர்புகொண்டு அந்தக் குழாய் வழியே பாய்ந்த நீரின் அளவை கேட்டறியும். காரணம் t1 என்னும் திறப்பி நுகரும் நீரின் அளவு p1 என்னும் குழாயினால் தீர்மானிக்கப்படுகிறது. இவ்வாறு ஒரு நிரலானது இறுதி விடையைக் கணக்கிட ஒன்றையொன்று அழைக்கும் பொருள்களைக் கொண்டதாய் உள்ளது. ஒவ்வொரு பொருளும் தத்தம் பங்களிப்பை நல்குகின்றன. அனைத்து நிரல் கூறுகளின் ஒருங்கிணைந்த செயல்பாடு நிரலின் இறுதி விடையை வழங்குகின்றது. பொருள்கள், தரவுகளை உள்ளீடுகளாக அனுப்பித் தமக்குள்ளே தகவல் பரிமாற்றம் செய்துகொள்கின்றன.

இன்றைய நிரலாக்கத்தில் மாணவர்கள், மேலாளர்கள், வங்கிக் கணக்குகள் இன்னும் இதுபோன்று நடப்புலகில் நாம் காணும் பொருள்கள் எடுத்தாளப்படுகின்றன. மிக இயல்பான முறையில் நடப்புலகப் பொருள்கள், நிரல்களில் கையாளப்படும் பொருள்களில் விளக்கம் பெறுகின்றன. இவ்வாறு ஒரு பொருளின் பண்பியல்பைச் செயற்கூறுகளாகப் பாவிக்கும் கருத்தோட்டமே பொருள்நோக்கு நிரலாக்கத்தின் குறிப்பிட்டுச் சொல்ல வேண்டிய சிறப்புக் கூறாகும்.

பயிற்சி வினாக்கள்

I. கோடிட்ட இடத்தை நிரப்புக.

அ) நடப்புலகப் பொருள்களைக் கணிப்பொறி நிரலில் எடுத்தாள ---------கருத்துரு உதவுகிறது.

ஆ) தரவுகளையும் அவற்றைக் கையாளும் செயற்கூறுகளையும் ஒன்றாகப் பிணைத்துவைக்கும் செயல்நுட்பம் ------------என்றழைக்கப்படுகிறது.

இ) வேறுபட்ட செய்திகளுக்கு மாறுபட்டுச் செயல்படும் ஒரு பொருளின் திறனே ------ என்றழைக்கப்படுகிறது.

ஈ) ஏற்கெனவே இருக்கும் தரவினத்தை அடிப்படையாகக் கொண்டு புதிய தரவினங்களை உருவாக்கும் செயல்நுட்பம் ------என்றழைக்கப்படுகிறது.

II. கீழ்க்காணும் வினாக்களுக்குச் சுருக்கமாக விடை அளிக்கவும்.

1. **பொருள்** என்பதன் உட்கருத்து என்ன ?
2. உறைபொதியாக்கம் என்பது என்ன ?
3. பல்லுருவாக்கம் என்பது மரபுரிமத்திலிருந்து எவ்வாறு வேறுபடுகிறது ?

III. கீழ்க்காணும் திட்டப்பணிக்கான தரவினங்களை வடிவமைக்கவும்

ஒரு நிறுவனம் தன்னுடைய நடவடிக்கைகளுக்காக ஒருதரவு மாதிரியத்தை (Data Model) தயாரிக்க விரும்புகிறது. அந்நிறுவனம் தனது பணியாளர்கள் அனைவரின் தகவலையும் சேமித்து வைத்துள்ளது. பணியாளர்களின் பொதுவான விவரங்கள் : Name, Date of Birth, Language, Nativity.

பணியாளர்களின் பணிப்பொறுப்பு அடிப்படையில் சேமித்து வைக்கப்பட்டுள்ள கூடுதல் விவரங்கள்

அ)	Stores	-	Date of joining, Dept, Salary
ஆ)	Scientist	-	Area of specialization, Current Project Details, Paper Presentations
இ)	Technician	-	Height, weight, ailments, risk factor, department, Wages

பாடம் 11

சி++ மொழியின் முன்னோட்டம்

(Overview of C++)

11.1 முன்னுரை

சி++ மொழி 1980களின் தொடக்கத்தில் ஏடீ & டீ பெல் ஆய்வுக்கூடத்தில் **ஜேர்ன் ஸ்ட்ரௌஸ்ட்ரப்** (Bjarne Stroustrup) அவர்களால் உருவாக்கப்பட்டது. சி++ என்னும் பெயரைச் (சி பிளஸ் பிளஸ் என உச்சரிக்க வேண்டும்) சூட்டியவர் ரிக் மாஸ்சிட்டி என்பவர் ஆவார். ++ என்பது சி மொழியின் மிகுப்புச் செயற்குறி ஆகும்.

11.2 சி++ குறியுருத் தொகுதி (C++ Character Set)

சி+ மொழியைப் போன்றே சி++ மொழியும் ஒரு குறியுருத் தொகுதியைக் கொண்டுள்ளது. அதிலிருந்து **வில்லைகள்** (Tokens - நிரலாக்கக் குறிமுறைக்குத் தேவையான உறுப்புகளின் அடிப்படை வகைகள்) உருவாக்கப்படுகின்றன. குறியுருத் தொகுதிகளில் கீழ்க்காணும் குறியுருக்கள் அடங்கியுள்ளன:

A...... Z, a........z, 0 9,

+, -, /, *,\, (,), [,], {, },

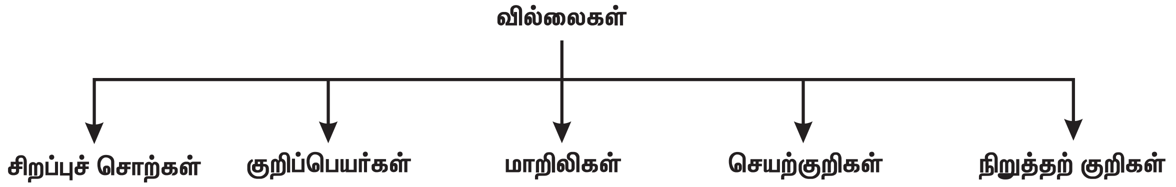
=, !=, <, >, . , ’ “ ; :

%, ! , &, ?, _, #, <=, >=, @,

வெற்று இடவெளி, கிடைமட்டத் தத்தல்

செலுத்தி திருப்பல் மற்றும் பிற குறியுருக்கள்

அடிப்படை இனங்கள் பற்றிச் சுருக்கமாகக் காண்போம். அடிப்படை இனங்கள் ஒட்டுமொத்தமாக வில்லைகள் (tokens) எனப்படுகின்றன. வில்லை என்பது ஒரு நிரலில் உள்ள மீச்சிறு தனித்த அலகு ஆகும். வில்லைகள் படம் 11.1–ல் உள்ளவாறு வகைப்படுத்தப்படுகின்றன.



படம் 11.1 வில்லைகளின் வகைப்பாடுகள்

11.2.1 சிறப்புச் சொற்கள்

ஒரு கணிப்பொறி மொழியின் நிரல்பெயர்ப்பிக்குப் (Language Compiler) புரிகின்ற பொருள்கொண்ட சொற்கள் சிறப்புச் சொற்கள் எனப்படுகின்றன. சிறப்புப் பயன்பாட்டுக்கென ஒதுக்கப்பட்ட சொற்கள் இவை. இச்சொற்களைச் சாதாரண குறிப்பெயர்களாகப் பயன்படுத்த முடியாது. சி++ மொழியில் பயன்படுத்தப்படும் சிறப்புச் சொற்களின் பட்டியலை அட்டவணை 11.1 காண்க.

auto	break	case	const	class	continue
default	delete	do	else	enum	for
friend	goto	if	inline	new	operator
private	protected	public	return	signed	sizeof
static	struct	switch	this	unsigned	virtual
while					

அட்டவணை 11.1 சிறப்புச் சொற்கள்

11.2.2. குறிப்பெயர்கள் (Identifiers)

குறிப்பெயர்கள், **மாறிகள்** (Variables) என்றும் அழைக்கப்படுகின்றன. மாறிகள் என்பவை, மதிப்பு அல்லது மாறிலிகளைத் தாங்கியுள்ள நினைவகப் பெட்டிகளைக் குறிக்கின்றன. மாறியின் பெயர் ஓர் எழுத்தில் அல்லது அடிக்கீறில் (Underscore) தொடங்கி, அதனைத் தொடர்ந்து எழுத்து அல்லது எண்களைக் கொண்டிருக்கும். எடுத்துக்காட்டாக _test; test; sum12 ஆகியவை ஏற்கத்தகு குறிப்பெயர்களாகும். தரவு இனங்களைப் பற்றிப் படித்த பிறகு மாறிகளைப் பற்றி மேலும் பார்ப்போம்.

11.2.3. மாறிலிகள் (Constants)

மாறிலி என்பது மாற்ற முடியாத மதிப்பைக் கொண்ட ஒரு தரவு விவரம் ஆகும். மாறிலி என்பது எண்வகையைச் சேர்ந்ததாகவோ, எண் அல்லாத வகையைச் சார்ந்ததாகவோ இருக்கலாம். எண் வகை மாறிலி எண் மதிப்புகளையே கொண்டிருக்கும். முழு எண்ணாக இருக்கலாம். பதின்மப் புள்ளி எண்ணாகவும் (decimal numbers) இருக்கலாம். முழு எண்கள் (Integers) மற்றும் மிதப்புப் புள்ளி (floating point) எண்கள் எண்வகை மாறிலிகள் ஆகும்.

11.2.4 முழு எண் மாறிலி (Integer Constants)

முழுஎண் மாறிலி என்பது குறைந்தது ஓர் இலக்கத்தையாவது கொண்டிருக்க வேண்டும். பின்னப்பகுதி எதுவும் இருக்கக்கூடாது.

- + அல்லது – என்கிற முன்னொட்டுக் குறி (prefix) இருக்கலாம்.
- **0 –** வில் தொடங்கும் எண்கள் எண்ம (Octal) மாறிலிகளாகக் கருதப்படும். (எ–டு) 010 = 8 ($[8]_{10} = [10]_{8}$)
- 0X எனத் தொடங்கும் எண்கள் பதினறும (hexadecimal) எண்களாகக் கருதப்படும் (எ.டு.) Ex. 0XF = 15 ($[15]_{10} = [F]_{16}$)

11.2.5. மிதப்புப் புள்ளி மாறிலி (Floating Point Constant)

மிதப்புப் புள்ளி என்பது குறியிட்ட மெய் எண்ணைக் குறிக்கும். அது, முழுஎண் பகுதி, பதின்மப் புள்ளி, பின்னப் பகுதி, அடுக்கெண் பகுதி ஆகிய நான்கு பகுதிகளைக் கொண்டிருக்கும். ஒரு மிதப்புப் புள்ளி எண் மாறிலியைக் குறிப்பிடும் போது முழுஎண் பகுதி, பின்னப் பகுதி ஆகிய இரண்டில் ஒன்றை விட்டுவிடலாம். ஆனால் இரண்டும் இல்லாமல் இருக்க முடியாது. எடுத்துக்காட்டாக **58.64** என்பது ஏற்கத்தகு மிதப்புப் புள்ளி எண் (மெய்யெண்) ஆகும். இந்த எண்ணை அடுக்கெண் முறையில் கீழ்க்காணுமாறு குறிப்பிடலாம்:

• 5.864E1 => 5.864 * 10^1 => 58.64

• 5.864E-2 => 5864 * 10^{-2} => 58.64

• 0.5864E2 => 0.5864*10^2 => 58.64

மிதப்புப் புள்ளி எண்களை அடுக்கெண் முறையில் குறிப்பிடும் போது E அல்லது e என்னும் எழுத்தைப் பயன்படுத்த வேண்டும்.

11.2.6 குறியுரு மாறிலி (Character Constant)

குறியுரு மாறிலி என்பது ஒற்றை மேற்கோள் குறிகளுக்குள் தரப்படும் ஒற்றை குறியுருவைக் கொண்டிருக்கும். சி++ மொழியின் குறியுருத் தொகுதியில் வரையறுக்கப்பட்டுள்ள எந்தக் குறியுருவாகவும் (ASCII தொகுதியின் அங்கமான எண்கள் கணித /ஒப்பீட்டு / பிற சிறப்புக் குறியுருவாகவும்) இருக்கலாம். தத்தல் (tab), பின்னிடவெளி (backspace) வரிச்செலுத்தி (line feed), மதிப்பிலி (null), பின்சாய்வு (backslash) ஆகிய சில குறியுருக்கள் **விடுபடு வரிசையாக** (escape sequence) குறிப்பிடப்படுகின்றன. விடுபடு வரிசைகள் பின்சாய்வுக் கோட்டை முன்னொட்டாகக் கொண்டு ஒற்றை எழுத்தால் குறிப்பிடப்படும். அட்டவணை 11.2–ல் விடுபடு வரிசைக் குறியுருக்களை காண்க.

விடுபடு வரிசை	வடிவற்ற குறியுரு
\a	மணி ஒலிப்பு
\b	பின்னிடவெளி
\n	புதிய வரி / வரிச்செலுத்தி
\t	கிடைமட்டத் தத்தல்
\v	செங்குத்துத் தத்தல்
\\	பின்சாய்வுக் கோடு
\’ (அ) \”	ஒற்றை/இரட்டை மேற்கோள்
\o	எண்ம எண்
\x	பதினறும எண்
\0	மதிப்பிலி

அட்டவணை 11.2 விடுபடு வரிசைக் குறியுருக்கள்

11.2.6. சர நிலையுரு (String Literal)

சர நிலையுரு என்பது இரட்டை மேற்கோள் குறிகளுக்குள் தரப்படும் குறியுருக்களின் வரிசையைக் கொண்டிருக்கும். சர நிலையுருக்கள் குறியுருக்களின் அணியாகக் (array of characters) கருதப்படுகின்றன; ஒவ்வொரு சர நிலையுருவும் தானமைவாக (by default) '\0' என்னும் சிறப்புக் குறியுருவை ஈற்றில் இணைத்துக் கொள்ளும். இது சரத்தின் இறுதியைக் குறிக்கிறது (எ.டு.) "Testing".

11.2.7. செயற்குறி (Operator)

செயற்குறி என்பது ஒரு மதிப்பை விடையாகக் பெறுவதற்கு நிகழ்த்தப்பட வேண்டிய ஒரு செயல்பாட்டைக் குறிக்கிறது, செயற்குறியானது **செயலேற்பிகளின்** (Operands) மீது செயல்படுகின்றது.

எடுத்துக்காட்டு:

RESULT = NUM1 + NUM2

இதில், NUM1, NUM2 ஆகியவை செயலேற்பிகள், + என்பது இரண்டு எண்களைக் கூட்டி விடைதரும் கூட்டல் செயற்குறி ஆகும். = என்னும் மதிப்பிருத்தல் (assignment) செயற்குறியின் உதவியுடன் விடை மதிப்பு, RESULT என்னும் மாறியில் சேமிக்கப்படுகிறது. சி++ மொழியில் பயன்படுத்தப்படும் செயற்குறிகளின் பட்டியலை அட்டவணை 11.3ல் காண்க.

[]	*	%	==	=	>=
( )	+	<<	!=	*=	&=
.	-	>>	^	/=	^=
->	~	<	|	+=	|=
++	!	>	&&	-=	, --
	size of	<=	||	% =	#
&	/	>=	?:	<<=	# #

அட்டவணை 11.3 சி++ மொழியின் செயற்குறிகள்

கீழ்க்காணும் செயற்குறிகள் சி ++ மொழிக்கே உரித்தானவை:

::   .*   ->*

மற்றும் ## ஆகிய செயற்குறிகள் முன்செயலியால் (Preprocessor) பயன்படுத்திக் கொள்ளப்படுபவை.

செயற்குறிகளை இவ்வாறு வகைப்படுத்தலாம்.

- கணக்கீடு (Arithmatic)
- மதிப்பிருத்தல் (Assignment)
- பொருட்கூறுத் தெரிவு (Component Selection)
- நிபந்தனை (Conditional)
- தருக்கம் (Logical)
- கையாளுகை (Manipulator)
- உறுப்பினர் குறிப்பு விலக்கம் (Member dereferencing)
- நினைவக மேலாண்மை (Memory Management)
- முன்செயலி (Preprocessor)
- ஒப்பீடு (Relational)
- வரையெல்லைக் கூறுபாடு (Scope Resolution)
- நகர்வு (Shift)
- இனமாற்றம் (Type Cast)

இடம்பெறும் செயலேற்பிகளின் எண்ணிக்கை அடிப்படையில் செயற்குறிகளை ஒருமம் (Unary) இருமம் (Binary), மும்மம் (Ternary) என மூவகையாகப் பிரிக்கலாம்.

எடுத்துக்காட்டு.

ஒருமச் செயற்குறிக்கு ஒரு செயலேற்பியே தேவை இருமச் செயற்குறிக்கு இரு செயலேற்பிகள் தேவை மும்மச் செயற்குறிக்கு மூன்று செயலேற்பிகள் தேவை

செயற்குறி	விளக்கம்
&	முகவரி சுட்டல்
!	தருக்க எதிர்மறை
*	உள்நோக்கல்
++	மிகுப்பு
~	பிட்நிலை நிரப்பு
_	குறைப்பு
-	ஒருமஎதிர்மம்
+	ஒரும நேர்மம்

படம் 11.4 (அ) ஒருமச் செயற்குறிகள்

வகை	செயற்குறி	விளக்கம்
கூட்டல்	+	இருமக் கூட்டல்
	-	இருமக் கழித்தல்
பெருக்கல்	*	பெருக்கல்
	/	வகுத்தல்
	%	வகுமீதி
நகர்வு	<<	இடது நகர்வு
	>>	வலது நகர்வு
பிட்நிலை	&	உம்
	|	அல்லது
	^	தனித்த அல்லது
தருக்கம்	&&	தருக்க உம்
	||	தருக்க அல்லது
மதிப்பிருத்தல்	=	மதிப்பிருத்தல்
	/=	வகுத்து மதிப்பிருத்தல்
	+=	கூட்டி மதிப்பிருத்தல்
	*=	பெருக்கி மதிப்பிருத்தல்
	%=	வகுமீதி மதிப்பிருத்தல்
	-=	கழித்து மதிப்பிருத்தல்
	<<=	இடம்நகர்ந்து மதிப்பிருத்தல்
	>>=	வலம்நகர்ந்து மதிப்பிருத்தல்
	&=	பிட்நிலை உம் மதிப்பிருத்தல்
	|=	பிட்நிலை அல்லது மதிப்பிருத்தல்
ஒப்பீடு	<	விடச்சிறிது
	>	விடப்பெரிது
	<=	விடச்சிறிது அல்லது நிகர்
	>=	விடப்பெரிது அல்லது நிகர்
நிகர்ப்பாடு	==	நிகர்
	!=	நிகரில்லை
பொருட்கூறு தெரிவு	"	நேரடிப் பொருட்கூறு தெரிவு
	->	மறைமுகப்பொருட்கூறுதெரிவு
இனக்குழு உறுப்பினர்	::	வரையெல்லை அணுகல் / கூறுபாட்டுச் செயற்குறி
	.*	குறிப்பு விலக்குச் செயற்குறி
நிபந்தனை	->*	இனக்குழு உறுப்புக்கான குறிப்பு விலக்கம்
	?:	மும்மச் செயற்குறி
காற்புள்ளி	,	மதிப்பிடல்

அட்டவணை 11.4 (ஆ) இருமச் செயற்குறிகள்

11.2.7.1 கணக்கீட்டுச் செயற்குறிகள் (Arithmatic Operators)

கணக்கீட்டுச் செயற்குறிகள் கணிதச் செயல்பாடுகளை நிறைவேற்றப் பயன்படுத்தப்படுகின்றன. கணக்கீட்டுச் செயற்குறிகளின் பட்டியல் காண்க.

செயற்குறி	விளக்கம்
+	கூட்டல்
-	கழித்தல்
*	பெருக்கல்
/	வகுத்தல்
%	முழுஎண் வகுமீதி
+=, -= *=, /= %=	கணக்கிட்டு மதிப்பிருத்தல்

கணக்கீட்டுச் செயற்குறிகள், எண்வகை மாறிலிகள்/மாறிகள், கணக்கீட்டுச் செயற்குறிகளால் இணைக்கப்பட்ட செயற்கூறு அழைப்பு ஆகியவற்றை உள்ளடக்கி, கணக்கீட்டுக் கோவைகள் (arithmetic expressions) அமைக்கப்படுகின்றன.

எடுத்துக்காட்டுகள்:

- a = -5;
- a = +b;
- a /= 5; (a = a/5)
- a++; (பின் மிகுப்புச் செயற்குறி. a=a+1 என்பதற்கு நிகர்)
- a-- ; (பின் குறைப்புச் செயற்குறி. a=a-1 என்பதற்கு நிகர்)
- ++a; (முன் மிகுப்புச் செயற்குறி. a=a+1 என்பதற்கு நிகர்)
- -- a ; (முன் குறைப்புச் செயற்குறி. a=a-1 என்பதற்கு நிகர்)
- a *= 2 (a = a * 2)
- a %= 5 (a = a % 5)
- a = b + pow(x,y); ($a = b + x^{y}$)

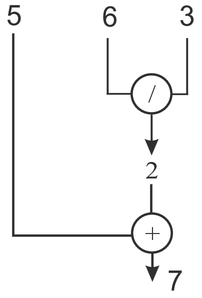
செயற்குறிகள் முன்னுரிமையின் அடிப்படையில் நிறைவேற்றப்படுகின்றன. செயலேற்பிகளும், செயற்குறிகளும் குறிப்பிட்ட தருக்க முறைப்படி குழுவாக்கப்பட்டு மதிப்பிடப்படுகின்றன. இத்தகைய தருக்கக் குழுவாக்கம் **தொடர்புறுத்தம்** (association) எனப்படுகிறது. **திசைமுகம்** என்றும் கூறலாம். கணிதச் செயற்குறிகளின் வகை மற்றும் தொடர்புறுத்தத்தை அட்டவணை 11.6 ல் காண்க.

செயற்குறி முன்னுரிமை	**வகை**	**திசைமுகம்**
(), [ ]	கணிதம்	இடமிருந்து வலம்
பின்னொட்டு ++, -- ,	கணிதம்–ஒருமம்	இடமிருந்து வலம்
முன்னொட்டு ++, -- ஒரும +, ஒரும –	கணிதம்–ஒருமம்	வலமிருந்து வலம்
*, / , %	கணிதம்–இருமம்	இடமிருந்து வலம்
+ , -	கணிதம்–இருமம்	இடமிருந்து வலம்

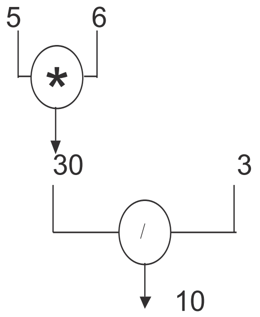
அட்டவணை 11.6 கணிதச் செயற்குறிகளின் முன்னுரிமை

கீழ்க்காணும் எடுத்துக்காட்டுகள், கணக்கீட்டுக் கோவைகள் மதிப்பிடப்படும் வரிசைமுறையை விளக்குகின்றன.

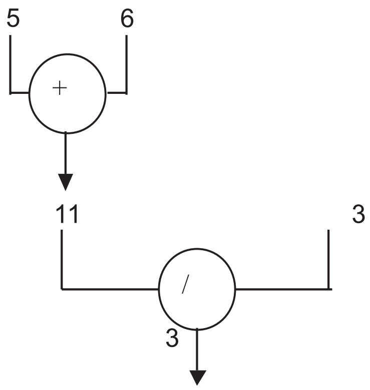
5 + 6/3 = 7



5 * 6 / 3 = 10

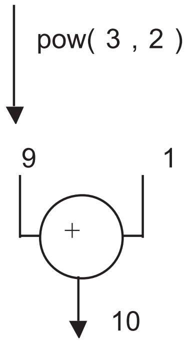


(5 + 6) / 3= 3

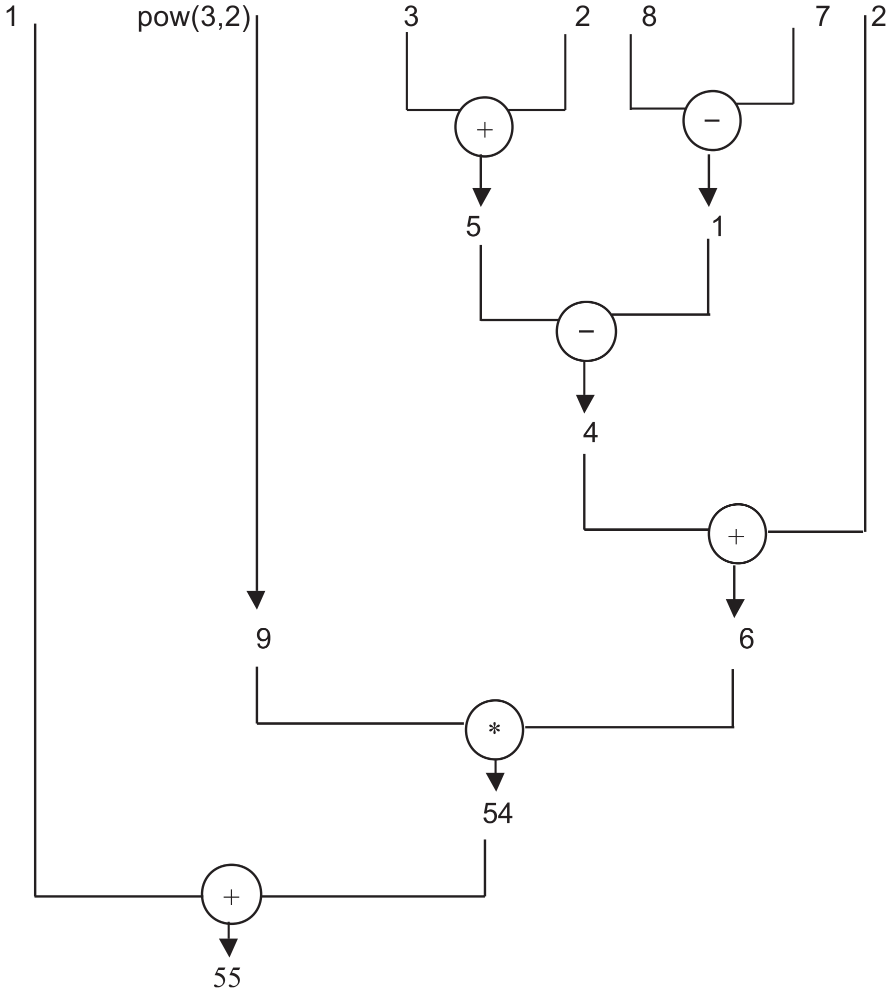


அனைத்து உள்ளீடுகளும் **int** இனம் என்பதால் விடை **3** ஆகும். ஏதேனம் ஓர் உள்ளீடு float இனம் எனில், விடை **3.66** எனக் கிடைக்கும்.

1 + pow (3 , 2)=10



1 + pow (3,2) * [(3+2) - (8-7) + 2] = 55



மிகுப்பு, குறைப்புச் செயற்குறிகள் சி மொழியில் உள்ளவாறே சி++ மொழியிலும் செயல்படுத்தப்படுகின்றன. இச்செயற்குறிகளைக் கொண்ட கோவைகள், மதிப்பிடப்படும் முறைகளை அட்டவணை **11.7** ல் காண்க.

கோவை	செயல்பாடு	எடுத்துக்காட்டு

a++	a- ன் மதிப்பைப் பெற்று அந்த மதிப்பில் 1 மிகுக்கவும்	a=5;
		c=a++;
		செயலாக்கம்:
		c=a;
		a=a+1;
		எனவே, c என்னும் மாறியில் இருத்தப்படும் மதிப்பு 5 ஆகும்

++a	a-ன் மதிப்பில் 1 மிகுத்து அந்த மதிப்பைப் பெறுக	a=5;
		c=++ a;
		செயலாக்கம்:
		a=a+1;
		c=a;
		எனவே, உ என்னும் மாறியில் இருத்தப்படும மதிப்பு 6 ஆகும்

a--	a-ன் மதிப்பைப் பெற்று அந்த மதிப்பில் 1 குறைக்கவும்	a=5;
		c=a--;
		செயலாக்கம்:
		c=a;
		a=a -1;
		இப்போது c–ன் மதிப்பு என்னவாக இருக்கும் ?

--a	a-ன் மதிப்பில் ஒன்றைக் குறைத்து அந்த மதிப்பைப் பெறுக	a=5;
		c=- - a;
		செயலாக்கம்:
		a=a-1;
		c=a;
		இப்போது c–ன் மதிப்பு என்னவாக இருக்கும் ?

அட்டவணை 11.6 மிகுப்பு, குறைப்புச் செயற்குறிகள்

அட்டவணை 11.7–ல் தரப்பட்டுள்ள கட்டளைத் தொகுதிகளில் மாறிகளில் இருத்தப்படும் மதிப்புகள் என்னவாக இருக்கும் ?

1.a=5;
b=5;
a=a+b++;
a மாறியில் இருத்தப்படும் மதிப்பு––––––
	2.x=10;
	f=20;
	c=x+++++f;
	மாறியில் இருத்தப்படும் மதிப்புகள்
	c–ல்––––
	x –ல்––––
	f–ல் –––––
		3.fun=1;
		sim=2;
		final=--fun +++sim-fun- -
		மாறியில் இருத்தப்படும் மதிப்புகள்
		fun–ல்––––
		sim-ல்––––
		final–ல் –––––

அட்டவணை 11.7 எளிய வினாக்கள்

11.2.7.2. ஒப்பீட்டுச் செயற்குறிகள் (Relational Operators)

மதிப்புகளை ஒப்பிட ஒப்பீட்டுச் செயற்குறிகள் பயன்படுத்தப்படுகின்றன. ஒப்பீட்டுச் செயற்குறிகளின் பட்டியல்:

- = = நிகரானது
- > விடப்பெரிது
- < விடச்சிறிது
- >= விடப்பெரிது அல்லது நிகர்
- <= விடச்சிறிது அல்லது நிகர்
- != நிகரில்லை

ஒப்பீட்டுச் செயற்குறிகள் மதிப்புகளை ஒப்பிடப் பயன்படுத்தப்படுகின்றன. இரண்டு செயலேற்பிகள் ஓர் ஒப்பீட்டு செயற்குறியால் இணைக்கப்பட்டு ஒப்பீட்டுக் கோவைகள் அமைக்கப்படுகின்றன. எடுத்துக்காட்டாக, கீழ்க்காணும் நிபந்தனைகளை அமைக்க ஒப்பீட்டுச் செயற்குறிகள் பயன்பட்டுள்ளன.

- 10 > 20
- 500.45 <= 1005
- 99 != 99.5
- 9 = = 9

ஒப்பீட்டுச் செயல்பாடு, **சரி** (true) அல்லது **தவறு** (false) என்கிற விடையைத் தரும். எண் மாறிலி 0 (சுழியம்) தவறு (false) என்னும் மதிப்பைக் குறிக்கிறது. சுழியம் அல்லாத எந்த எண் மதிப்பும் சரி (true) என்பதைக் குறிக்கும். மேற்கண்ட கோவைகளின் வெளியீடு இவ்வாறு இருக்கும்.

- 0 (10 > 20 என்பது தவறு)
- 1 (500.45 < 1005 என்பது சரி. எனவே ஒரு சுழியமில்லா மதிப்பு)
- 1 (99 != 99.5 5 என்பது சரி. எனவே ஒரு சுழியமில்லா மதிப்பு)
- 1 (9 = = 9 என்பது சரி. எனவே ஒரு சுழியமில்லா மதிப்பு)

கீழ்க்காணும் கோவையின் மதிப்பு என்னவாக இருக்கும் **?**

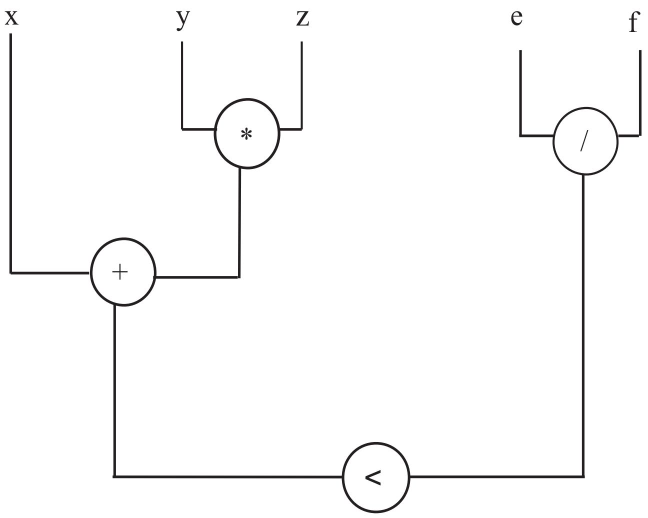
num1=99, num2=20, num 3=10 எனில்
(num1+num2 - num 3)/5 * 2<(num1%10) எவ்வளவு **?**

அட்டவணை 11.8–ல் உள்ள ஒப்பீட்டுக் கோவைகளை மதிப்பிடுக.

செயற்குறி	**கோவை**	**விடை**
= =	5 = = 6	0
! =	'a' ! = 'a'	
>	5 > 6	
	'a' > 'A'	
<	5 < 6	
	'a' < 'A'	
>=	'a' >= 'z'	
	5 >= 5	
<=	'a' <= 'z'	
	5 <= 6	

அட்டவணை 11.8 ஒப்பீட்டுக் கோவைகளை மதிப்பிடுக

ஒப்பீட்டுச் செயற்குறிகள், கணக்கீட்டுச் செயற்குறிகளை விடக் குறைந்த முன்னுரிமை உடையவை. எடுத்துக்காட்டாக, x + y * z < e / f என்னும் கோவை இவ்வாறு மதிப்பிடப்படும்.



11.2.7.3 தருக்கச் செயற்குறிகள் (பூலியன் செயற்குறிகள்)

ஒன்று அல்லது ஒன்றுக்கு மேற்பட்ட நிபந்தனைகளின் விடைகளை தருக்கச் செயற்குறிகள் ஒன்றிணைக்கின்றன. &&(AND), || (OR) ! (NOT) - ஆகியவை தருக்கச் செயற்குறிகள் ஆகும்.

எடுத்துக்காட்டு:

c = 5 , d = 6 , choice = 'y', term = '2' **(சரி** என்பதை **1** எனவும், **தவறு** என்பதை **0** எனவும் கருதிக் கொள்க**)**

Result_1 = (c = = d) && (choice != term)

Result_2 = ('y' = = 'Y') || (term != '0')

Result_3 = (c = = d) && ('y' = = 'Y')|| choice != term)

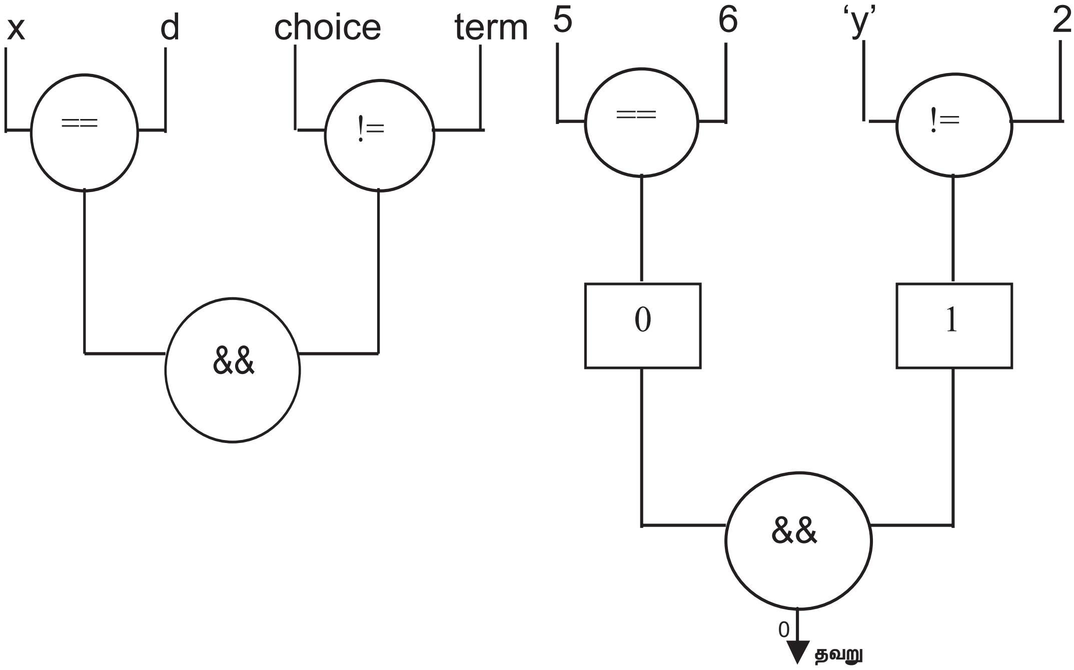
Result_4 = (c = = d) || ('y' = = 'Y') && choice ! = term)

Result_1, Result_2 ஆகியவற்றில் இருத்தப்பட்ட மதிப்பு யாது **?**

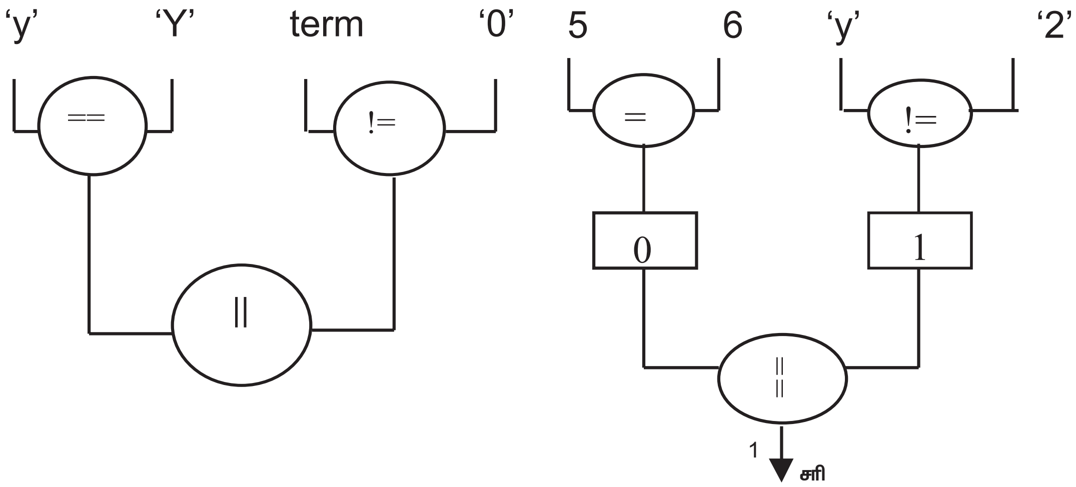
Result_1ல் உள்ள மதிப்பு **0** (தவறு) Result_2ல் உள்ள மதிப்பு **1** (சரி)

Result_3ல் உள்ள மதிப்பு **1** (சரி); Result_4ல் உள்ள மதிப்பு **0** (தவறு)

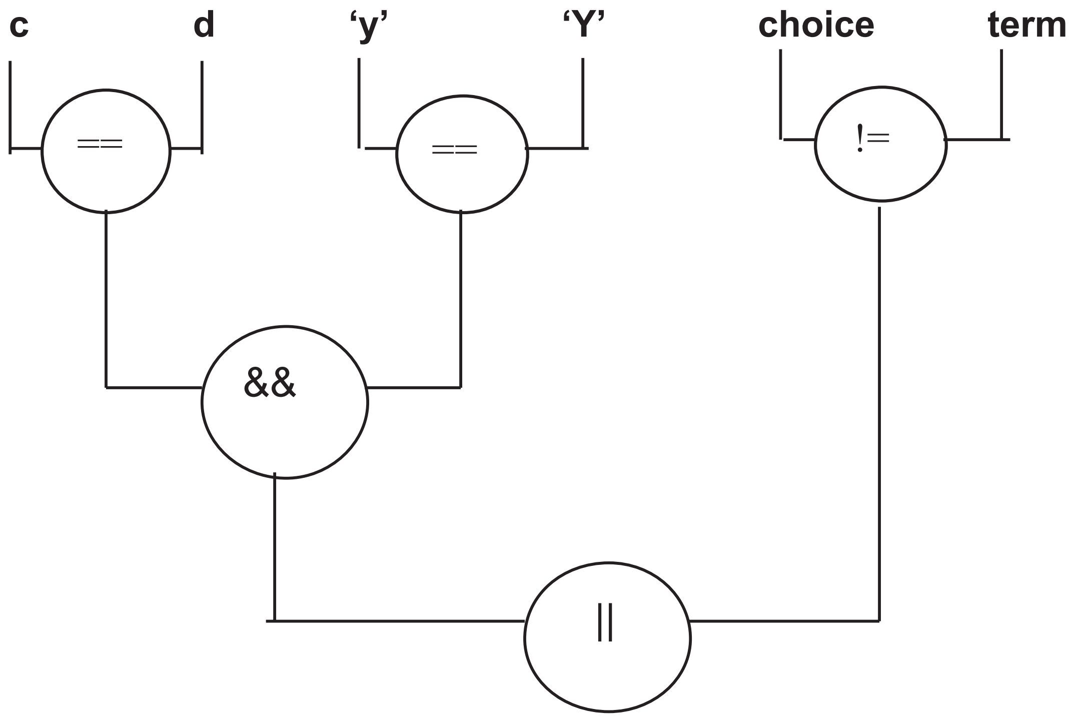
Result_1= (c==d) && (choice != term) மதிப்புகளைப் பதிலீடு செய்யின்



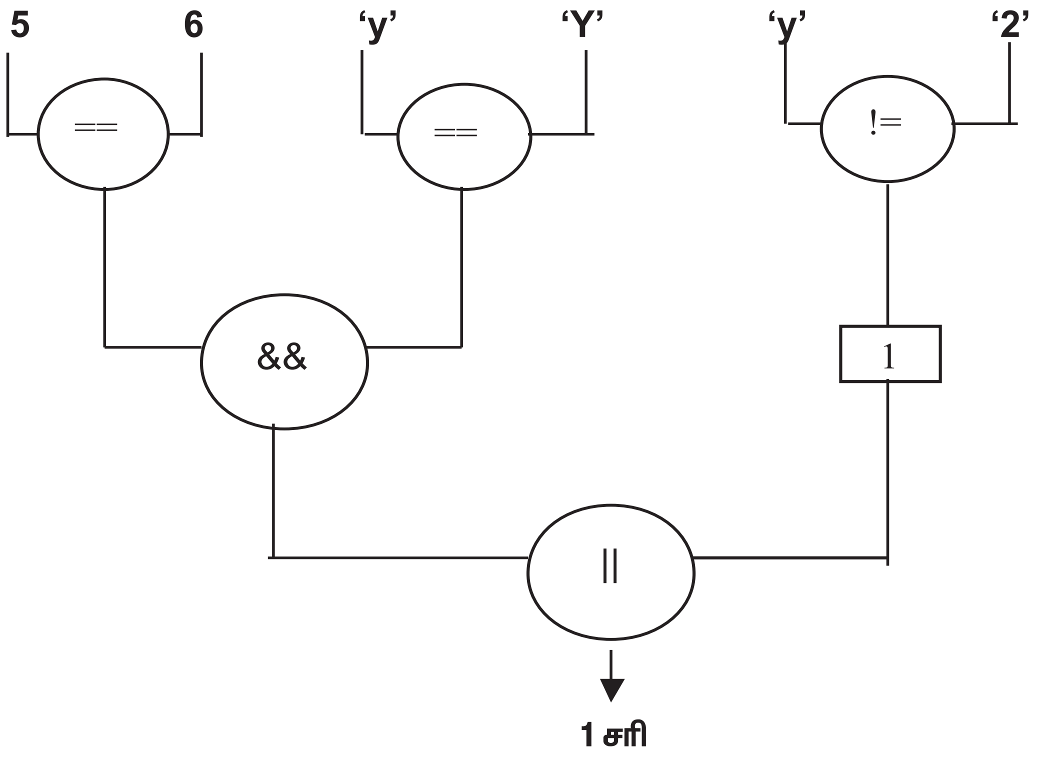
Result_2 = ('y'=='Y') || (term != 0) மதிப்புகளைப் பதிலீடு செய்யின்



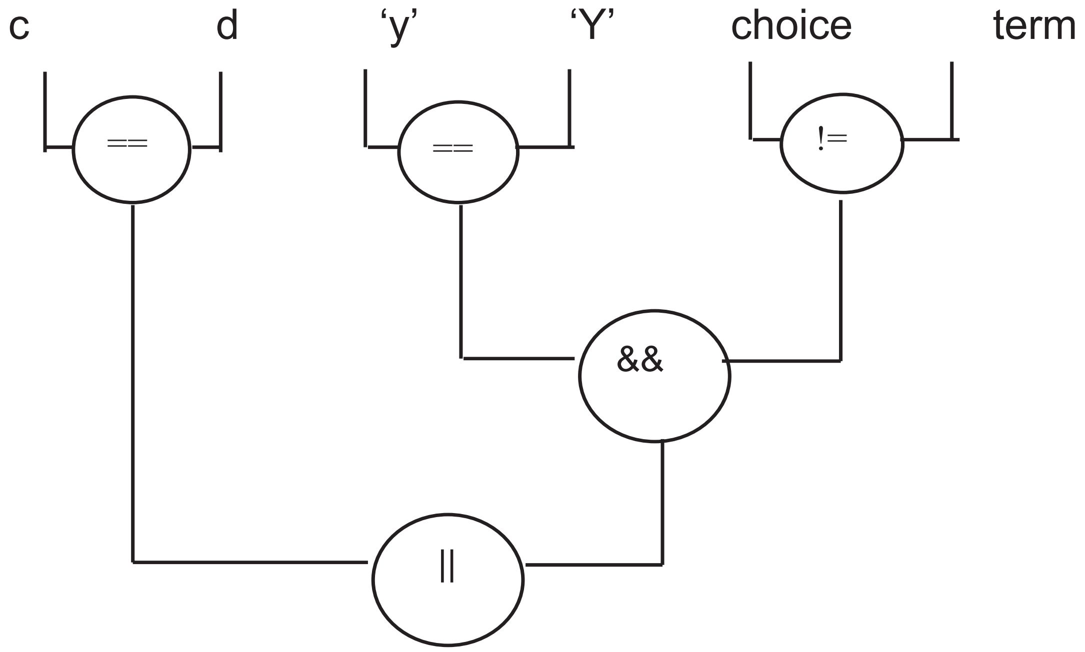
Result_3 = (c==d) && (‘y’ == ‘Y’) || (choice != term)



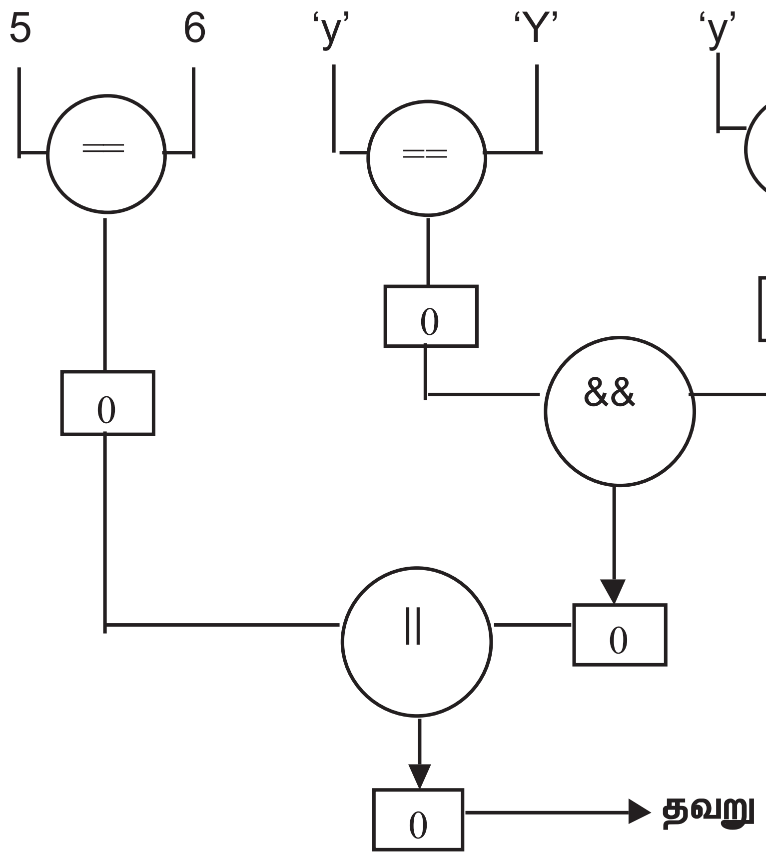
மதிப்புகளைப் பதிலீடு செய்யின்



Result_4 = (c == d) || (‘y’ == ‘Y’) && (choice != term)

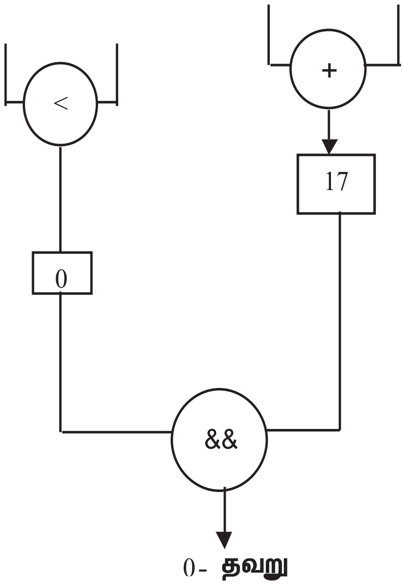


மதிப்புகளைப் பதிலீடு செய்யின்



தருக்கச் செயற்குறிகள், ஒப்பீட்டுச் செயற்குறிகள் மற்றும் கணக்கீட்டுச் செயற்குறிகளைவிடக் குறைந்த முன்னுரிமை கொண்டவை. பின்வரும் கோவையின் மதிப்பைக் கண்டறிய முடியுமா **?**

5 < 4 && 8 + 9



11.2.7.4 நிபந்தனைக் செயற்குறி (?:) (Conditional Operator)

நிபந்தனைச் செயற்குறி,

(num1>num2)? "சரி": "தவறு"

என அமையும் **?:** என்பது மும்மச் செயற்குறி. (num1>num2), "சரி", "தவறு" –ஆகியவை செயலேற்பிகள். மும்மச் செயற்குறியை **நிபந்தனைச் செயற்குறி** என்றும் அழைக்கிறோம். இச்செயற்குறியின் பொதுவான கட்டளை அமைப்பு.

E1? E2: E3 ;

என அமையும். E1, E2, E3 ஆகியவை செயலேற்பிகள் ஆகும். E1 என்பது ஒரு நிபந்தனையாக இருக்கும். E2, E3 ஆகியவை மதிப்புகளாகவோ, கூற்றுகளாகவோ இருக்கலாம். எடுத்துக்காட்டாக, இரண்டு எண்களில் பெரிய எண்ணை ஒரு மாறியில் இருத்துவதற்கு,

max=(num1>num2)?num1:num2;

எனக் கட்டளை அமைக்கலாம். num1 என்னும் மதிப்பு num2 –ஐக் காட்டிலும் பெரிது எனில் max ல் num1 இருத்தப்படும். இல்லையேல், max-ல் num2 இருத்தப்படும்.

கீழேயுள்ள கட்டளைத் தொகுதியில், x என்னும் மாறியில் இருத்தப்படும் மதிப்பு என்னவென்று சொல்ல முடியுமா **?**

a = 10;

b = 10;

x = (a < b) ? a*a : b % a;

11.2.7.5 **மதிப்பிருத்து செயற்குறிகள்** (Assignment Operators)

= என்பது சாதாரண மதிப்பிருத்து செயற்குறி ஆகும். ஒரு கோவையின் (செயற்குறியின் வலப்பக்கம் இருக்கும்), விடையை ஒரு மாறியில் (செயற்குறியின் இடப்பக்கம் இருக்கும்) இருத்துவதற்கு இது பயன்படுத்தப்படுகிறது. இந்தச் சாதாரண மதிப்பிருத்து செயற்குறி தவிர வேறு பத்து குறுக்குவழி மதிப்பிருத்து செயற்குறிகளும் உள்ளன. அனைத்து மதிப்பிருத்து செயற்குறிகளையும் கீழேயுள்ள அட்டவணை 11.9ல் காண்க.

கோவை	**செயலாக்கம்**	**விடை**
A=5	A என்னும் மாறியில் 5 என்னும் மதிப்பு இருத்தப்படுகிறது	மாறி 5 என்னும் மதிப்பைப் பெறுகிறது
A+=2	A+=2 என்பதன் பொருள் A=A+2 ஆகும்	A–யில் இருத்தப்படும் மதிப்பு 7
A*=4	A=A*4	A–யில் இருத்தப்படும் மதிப்பு 20
A/=2	A=A / 2	A–யில் இருத்தப்படும் மதிப்பு 2
A-=2	A=A - 2	A–யில் இருத்தப்படும் மதிப்பு 3
A%=2	A= A%2	A–யில் இருத்தப்படும் மதிப்பு 1
a=5, b=6, c=7 எனில் கீழ்க்காணும் கோவைகளின் மதிப்பைக் காண்க		
a+=b*c c*=a+a/b b+=a%2*c		

அட்டவணை 11.9ல் மதிப்பிருத்து செயற்குறிகள்

சி++ மொழியில் பயன்படுத்தப்படும் அனைத்துச் செயற்குறிகளின் முன்னுரிமையையும் நிறைவேற்றப்படும் திசைமுகத்தையும் அட்டவணை 11.10ல் காண்க

செயற்குறி முன்னுரிமை	**வகை**	**திசைமுகம்**
( ) [ ]		இடமிருந்து வலம் இடமிருந்து வலம்
பின்னொட்டு++, – – முன்னொட்டு++, – – !–தருக்க இல்லை ஒரும +, ஒரும–	கணித ஒருமம் கணித ஒருமம் தருக்க ஒருமம் கணித ஒருமம்	வலமிருந்து இடம் வலமிருந்து இடம் வலமிருந்து இடம் இடமிருந்து வலம்
* / %	கணித இருமம்	இடமிருந்து வலம்
+ -	கணித இருமம்	இடமிருந்து வலம்

<table>
<tr><td>< <= > >=</td><td>ஒப்பீட்டு இருமம்</td><td>இடமிருந்து வலம்</td></tr>
<tr><td>==, ! =</td><td>ஒப்பீட்டு இருமம்</td><td>இடமிருந்து வலம்</td></tr>
<tr><td>&& (AND)</td><td>தருக்க இருமம்</td><td>இடமிருந்து வலம்</td></tr>
<tr><td>II (OR)</td><td>தருக்க மும்மம்</td><td>இடமிருந்து வலம்</td></tr>
<tr><td>? :</td><td>தருக்க மும்மம்</td><td>இடமிருந்து வலம்</td></tr>
<tr><td>= *= /= % = +=
-= <<= >>=
& = ^= |=</td><td>மதிப்பிருத்தல்</td><td>வலமிருந்து இடம்</td></tr>
</table>

அட்டவணை 11.10 செயற்குறி முன்னுரிமை

(குறிப்பு– C++ மொழிக்கே உரிய சிறப்புச் செயற்குறிகள் அந்தந்தப் பாடங்களில் விளக்கப்படும்)

11.2.8 நிறுத்தற் குறிகள் (Punctuators)

நிறுத்தற் குறிகள் என்பவை குறிப்பிட்ட பணியைச் செய்யும் குறியுருக்களாகும். நிறுத்தற் குறிகளையும் அவற்றின் பயன்பாட்டையும் அட்டவணை 11.11ல் காண்க.

நிறுத்தற் குறி	பயன்பாடு
;	ஒரு சி++ கட்டளையை முடித்து வைக்கிறது
//	குறிப்புரைகளுக்குப் (Comments) பயன்படுத்தப்படுகிறது
/* */	இவற்றில் உள்ளடங்கிய உரைத் தொகுதி குறிப்புரையாகக் கருதப்படும்
{ }	சி++ கட்டளைகளை ஒரு தொகுதியாகக் குறிக்கப் பயன்படுகிறது. ஒரு செயற்கூறின் கட்டளைத் தொகுதி இந்த அடைப்புக் குறிகளுக்குள் இருக்கும்
[ ]	ஓர் அணியின் குறிப்பிட்ட உறுப்பைச் சுட்டுகின்ற சுட்டெண் (Index) இந்த அடைப்புக் குறிகளுக்குள் இடம்பெறும்
‘ ’	ஒற்றைக் குறியுரு இந்த மேற்கோள் குறிகளுக்குள் தரப்படும்
“ ”	சரம் இந்த மேற்கோள் குறிகளுக்குள் இடம்பெறும்

அட்டவணை 11.11 நிறுத்தற் குறிகளும் அவற்றின் பயன்பாடும்

11.3 **தரவு இனங்கள்** (Data Types)

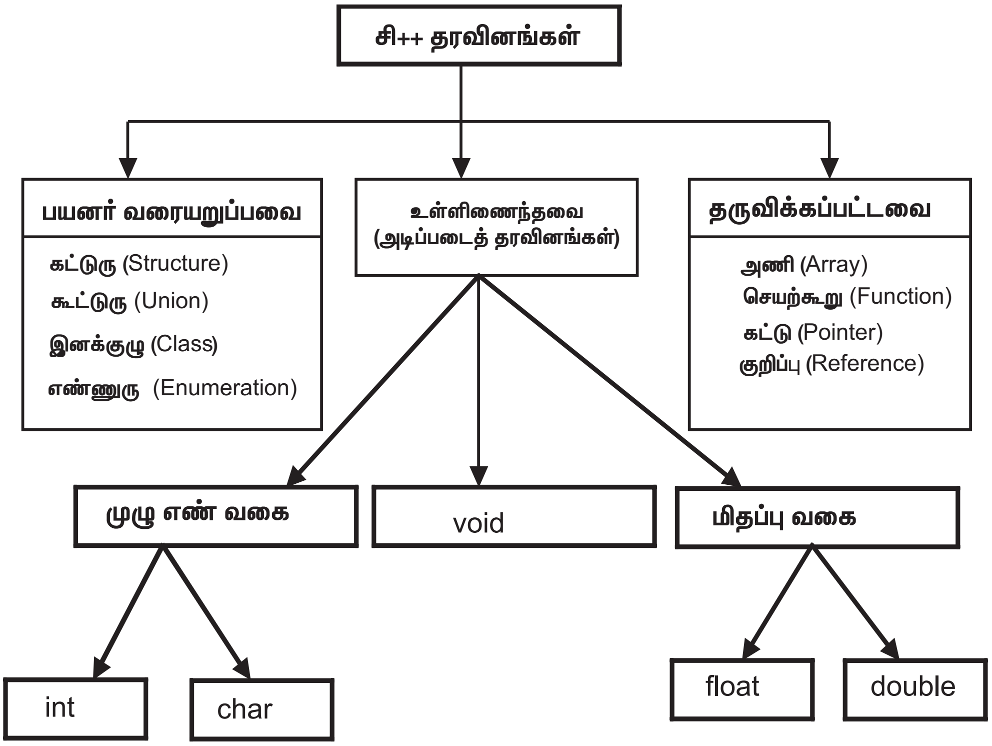
ஒரு நிரலாக்க மொழியில் கையாளப்படும் மாறிகள் ஒவ்வொன்றும் குறிப்பிட்ட வகை மதிப்புகளையே ஏற்கும். இத்தகைய தரவின் வகைகளையே **தரவு இனங்கள்** (Data Types) என்கிறோம். சி++ மொழியில் தரவுகளை வேறுபட்ட இனங்களாகப் பிரித்துக் காணும் திறன், சிக்கல் மிகுந்த பொருள்களை நிரல்களில் எடுத்தாளும் பணியை எளிமையாக்குகிறது. தரவுகளைப் பல்வேறு வகையினங்களாகப் பிரித்தமைப்பதற்கு இரண்டு காரணங்கள் உள்ளன.

- ஒவ்வொரு தரவினத்துக்கும் நிரல்பெயர்ப்பி (Complier) உரிய அகநிலை உருவகிப்பைப் பயன்படுத்திக் கொள்ளும்.
- நிரல்களை உருவாக்கும் நிரலர் ஒவ்வொரு தரவினத்துக்கும் ஏற்ற செயற்குறிகளைப் பயன்படுத்திக் கொள்ள முடியும்.

தரவினங்களை முப்பெரும் வகையினங்களாகப் பிரிக்கலாம்.

- பயனர் வரையறுக்கும் தரவினம் (User defined type)
- உள்ளிணைந்த தரவினம் (Built - in type)
- தருவிக்கப்பட்ட தரவினம் (Derived type)

மேற்கண்ட பரந்த வகைப்பாட்டினை உட்பிரிவுகளோடு படம் 11.2ல் காண்க:

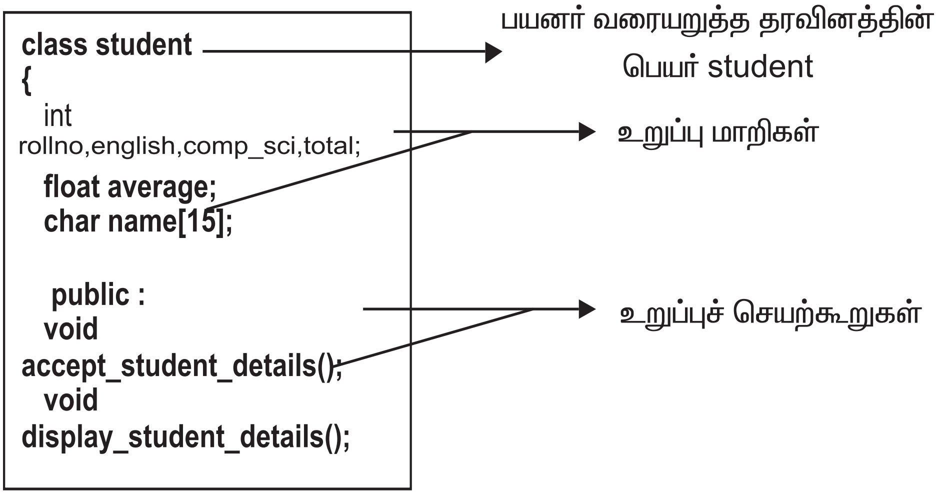


படம் 11.2 சி++ தரவினங்கள்

11.3.1. **பயனர் வரையறுக்கும் தரவினம்** (user defined data type)

நிரலர் தன்னுடைய சொந்தத் தரவினங்களையும், அவை ஏற்கின்ற மதிப்புகளையும் வரையறுத்துக் கொள்ள வகை செய்யப்பட்டுள்ளது. ஒரு நிரலின் படிப்பெளிமையை (readability) மேம்படுத்த இது உதவுகிறது.

எடுத்துக்காட்டாக, கீழேயுள்ள பயனர் வரையறுத்த தரவினத்தைக் காண்க.



படம் 11.3 பயனர் வரையறுத்த தரவினம்

பயனர் வரையறுத்த student என்னும் தரவினம் **இனக்குழு** (class) என்னும் தரவு வகையைச் சார்ந்தது. **உறுப்பு மாறிகள்**, **அவற்றோடு தொடர்புடைய செயற்கூறுகளின் வடிவில்** ஒரு மாணவரின் பரிமாணங்களை இது வரையறுத்துள்ளது. ஒரு மாணவரின் விவரங்களைப் பெறவும், அவ்விவரங்களைத் திரையில் காட்டவும் செயற்கூறுகள் வரையறுக்கப்பட்டுள்ளன. இவ்வாறு, class student என்னும் தரவினம் தேவையான தரவுகளையும் அவற்றோடு தொடர்புடைய செயற்கூறுகளையும் மாணவர்க்கான ஒரு புதிய தரவின வடிவில் ஒன்றாகப் பிணைத்து வைத்துள்ளதன் மூலம் நிரலின் **தரத்தையும் படிப்பெளிமையையும் மேம்படுத்தியுள்ளது.**

ஏற்கெனவே நிலவும் ஒரு தரவினத்துக்குப் பயனர் ஒரு புதிய பெயரைச் சூட்டிக்கொள்ள முடியும். **இனவரையறை** (Type definition) என்னும் வசதி பயனர், தரவினக் குறிப்பெயர்களை உருவாக்கிக் கொள்ள அனுமதிக்கிறது. அதற்கான கட்டளை அமைப்பு:

typedef <தரவினம்> <பயனர் வரையறுக்கும் தரவினக் குறிப்பெயர்>;

எடுத்துக்காட்டு:

typedef int **marks;**

typedef char **grade;**

marks, grade ஆகியவை பயனர் வரையறுத்த தரவினக் குறிப்பெயர்களாகும். பயனர்கள் int, char மாறிகளை இவ்வாறு வரையறுக்கலாம்.

marks eng_marks, math_marks;

grade eng_grade, math_grade;

பொருள் பொதிந்த தரவினக் குறிப்பெயர்களை உருவாக்கிக் கொள்ள typedef உதவுகிறது. அவை நிரலின் படிப்பெளிமையை மேம்படுத்துகின்றன.

பயனர்கள் வரையறுக்கும் இன்னொரு தரவினம் எண்ணுருத் தரவினமாகும். பயனர்கள் குறிப்பெயர்களின் பட்டியலை ஒரு தரவினமாக வரையறுத்துக் கொள்ள இத்தரவினம் உதவுகிறது. **int இனக் குறியீட்டு எண்வகை மாறிலி** என்றும் இதனை அழைப்பர்.

கட்டளை அமைப்பு:

enum< தரவினக் குறிப்பெயர்> (மதிப்பு1, மதிப்பு2................மதிப்பு n);

எடுத்துக்காட்டுகள்

enum **working_days** (Monday, Tuesday, Wednesday, Thursday, Friday);

enum **holidays** (Sunday, Saturday);

working_days,holidays ஆகிய குறிப்பெயர்கள் பயனர் வரையறுத்த தரவினங்கள் ஆகும். (Monday, Tuesday, ...) என்பது **எண்ணுரு மாறிலிகள்** அல்லது **எண்வகை மாறிலிகள்** என்று அழைக்கப்படும் மதிப்புகளைக் கொண்ட பட்டியலாகும்.

இந்த எண்ணுருத் தரவினத்தில் பயனர்கள் கீழ்க்காணுமாறு மாறிகளை அறிவிக்கலாம்.

enum<குறிப்பெயர்> மாறி1, மாறி2................மாறி n;

எடுத்துக்காட்டாக, working_days என்னும் தரவினத்தில் first_workingday, last_workingday என்னும் மாறிகளைக் கீழ்க்காணுமாறு அறிவிக்க முடியும்

working_days first_workingday, last_workingday;

இந்த மாறிகள் working_days இனத்தில் வரையறுக்கப்பட்டுள்ள மதிப்புகளில் ஒன்றை மட்டுமே ஏற்கும்.

first_workingday= Monday;

last_workingday=Friday;

குறிப்பெயர் மாறிலிகள் Monday, Tuesday, Wednesday........ஆகியவை அக நிலையில் நிரல்பெயர்ப்பியால் முன்னியல்பாக 0–வில் தொடங்கி 1, 2, 3,............என வரிசையான int மாறிலிகளாகவே கையாளப்படுகின்றன. எனவே, மேற்கண்ட மதிப்பிருத்தல் கட்டளைகளை இவ்வாறும் எழுதலாம்.

first_workingday=0;

last_workingday=4;

பயனர்கள், குறிப்பெயர் மாறிலிகளுக்கு வெளிப்படையாக வேறு எண் மதிப்புகளைக் குறிப்பிட்டு int மாறிலிகளை இவ்வாறு மறுவரையறை செய்து கொள்ள முடியும்.

enum working_days (Monday=1, Tuesday, Wednesday, Thursday, Friday);

இதில் Monday என்பது 1 என்ற மதிப்பைப் பெறும். அடுத்துள்ள மாறிலிகள் 2,3,4,........ என அடுத்தடுத்த int மாறிலி மதிப்புகளைப் பெறும்.

11.3.2. சேமிப்பு இனம் (Storage Class)

ஒரு மாறியின் அறிவிப்புக்கு முன்னால் **சேமிப்பு இனம்** என்னும் தகுதியாக்கியை (qualifier) இணைத்துக் கொள்ள முடியும். auto, static, extern, register என்னும் நான்கு வகையான சேமிப்பு இனங்கள் உள்ளன. static, register ஆகிய இன மாறிகளில் அவை அறிவிக்கப்படும்போதே 0 என்ற தொடக்க மதிப்பு தானாகவே இருத்தப்பட்டு விடுகிறது. auto இன மாறிகளில் அவ்வாறு தரவினத்தின் அடிப்படையில் ஏற்ற மதிப்புகள் தொடக்க மதிப்பாக இருத்தப்படுவதில்லை. auto மாறிகள், குப்பை (garbage) எனப்படும். வரையறுக்கப்படாத மதிப்புகளைப் பெறுகின்றன. சேமிப்பு இனங்களுக்கான பொருளும், பொருத்தமான எடுத்துக்காட்டுகளும் அட்டவணை 11.12 ல் வழங்கப்பட்டுள்ளன.

சேமிப்பு இனம்	பொருள்	எடுத்துக்காட்டு

auto	உள்ளக மாறிகள் (local variables) இவை அறிவிக்கப்படும் தொகுதிக்குள் மட்டுமே அறியப்படும். முன்னியல்பாகவே, உள்ளக மாறிகள் அனைத்தும் auto இனத்தைச் சார்ந்தவையே. எனவே auto என்னும் சேமிப்பு இனப்பெயர் பெரும்பாலும் குறிப்பிடப்படுவதில்லை.	void main()
		{
		auto float ratio ;
		auto int count;
		}
		ratio, count ஆகிய மாறிகள் main()செயற்கூறுக்குள் வரையறுக்கப்பட்டுள்ளன. இவை auto என்னும் சேமிப்பு இனத்தைச் சேர்ந்தவை

static	ஒரு செயற்கூறினுள் அல்லது தொகுதிக்குள் வரையறுக்கப்படும் மாறிகள் அந்த செய்றகூறு அல்லது தொகுதி செயல்பட்டு முடிந்தவுடன் அழிந்து போகின்றன. அவ்வாறின்றி மாறிகள் அவை அறிவிக்கப்படும் செயற்கூறு அல்லது தொகுதி செயல்பட்டு முடிந்தபின்னும் கணிப்பொறி நினைவகத்தில் தங்கியிருக்க static என்னும் பண்புணர்த்தி (Modifier) வகை செய்கிறது. அந்தமாறிகள் கடைசியாக இருத்தப்பட்ட மதிப்புகளைத் தக்கவைத்துக் கொள்கின்றன.	void fun()
		{
		static int x;
		x++;
		}

extern	இந்த வகையைச் சார்ந்த மாறிகள், நடப்பு நிரலில் அனைத்து செயற்கூறுகளிலும் பயன்படுத்தக்கூடிய முழுதளாவியமாறிகள் (global variables) ஆகும். இந்த மாறிகள் வேறொரு நிரலில் வரையறுக்கப்பட்டிருக்கும்.	extern int filemode; extern void factorial();

register	register என்னும் பண்புணர்த்தி அவ்வாறு அறிவிக்கப்படும் மாறிகளை சிபீயுவின் பதிவகங்களில் (registers) இருத்திவைக்குமாறு நிரல்பெயர்ப்பிக்கு உணர்த்துகிறது. இவற்றை மிக வேகமாய் அணுக முடியும்.	void fun()
		{
		register int i;
		}

அட்டவணை 11.12 சேமிப்பு இனக்குழுக்கள்

11.3.4. உள்ளிணைந்த தரவினங்கள் (Built in Data Types)

உள்ளிணைந்த தரவினங்கள் **மூலத் தரவினங்கள்** அல்லது **அடிப்படைத் தரவினங்கள்** என்றும் அழைக்கப்படுகின்றன. நிரல்பெயர்ப்பியில் அவை முன்வரையறுக்கப்பட்டுள்ளன. அடிப்படைத் தரவினங்களை முழுஎண் (integral), மிதப்பு(float), மதிப்பிலி (void) என மூன்று வகையாகப் பிரிக்கலாம்.

முழு எண் வகை, int, char ஆகிய தரவினங்களை உள்ளடக்கியது. int தரவினம். 1,2,3,.......என முழுஎண் மதிப்புகளையே ஏற்கும். பின்ன மதிப்புகளை ஏற்காது. char என்பது குறியுருத் தரவினம். எனினும் இது குறியுரு மதிப்பு, முழு எண் மதிப்பு ஆகிய இரண்டையுமே ஏற்கும். எடுத்துக்காட்டாக, ch என்னும் மாறியின் அறிவிப்பையும், அதில் தொடக்க மதிப்பிருத்தலையும் காண்க

Char ch = ‘A’;

இதே கட்டளையை,

Char ch = 65;

என்றும் அறிவிக்கலாம். (65 என்பது A என்னும் எழுத்தின் ஆஸ்க்கி மதிப்பாகும்). மேற்கண்ட இரண்டு கூற்றுகளும் ஒரே பணியையே நிறைவேற்றுகின்றன. ch என்னும் மாறியில் ‘A’ என்னும் எழுத்தை இருத்துகின்றன.

மிதப்புவகைத் தரவினம் float, double ஆகிய தரவினங்களை உள்ளடக்கியது. மிதப்புவகைத் தரவினம் பின்னப் பகுதியுடன் கூடிய எண் மதிப்புகளை இருத்திவைக்கும் திறன் படைத்தவை. (மிதப்புப் புள்ளி மாறிலிகள் பற்றி ஏற்கெனவே படித்துள்ளதை நினைவு கூர்க).

மதிப்பிலித் தரவினம் (void type) இரண்டு வகையில் பயன்படுகின்றன.

- ஒரு செயற்கூறு எந்த மதிப்பையும் திருப்பி அனுப்பாது என்பதைக் குறிக்க
- பொது இனச் சுட்டினை (generic pointer) அறிவிக்க

எடுத்துக்காட்டாக, கீழேயுள்ள void.cpp,fun.cpp ஆகிய சி++ நிரல்களில் வரையறுக்கப்பட்டுள்ள செயற்கூறுகளை நோக்குக.

```
Program  void.cpp
#include<iostream.h>
#include<conio.h>
void  fun(void)
{
  int a,b;
  cin >> a >> b;
  cout << a+b;
}

void main()
{
  fun();
}
```

```
Program  fun.cpp
#include<iostream.h>
#include<conio.h>
int fun(int a, int b)
{
  return  a+b;
}
void main()
{
   int a = 5 , b = 6, sum = 0;
   sum =  fun(a,b);
   cout << sum;
}
```

void.cpp நிரலில் void cpp (void) என அறிவிக்கப்பட்டுள்ள முன்வடிவு (prototype) இச்செயற்கூறு எவ்வித மதிப்பையும் திருப்பியனுப்பாது என்பதையும், அளபுருக்களின் (parameters) வடிவில் எவ்வித மதிப்புகளையும் ஏற்றுக் கொள்ளாது என்பதையும் உணர்த்துகிறது. எனவே தான் main ( ) செயற்கூறில், அழைப்புக் கூற்று fun ( ) என்று இடம் பெற்றுள்ளது. fun.cpp நிரலில் அறிவிக்கப்பட்டுள்ள int fun (int a, int b)என்னும் முன்வடிவு fun( ) செயற்கூறு ஓர் int மதிப்பைத் திருப்பியனுப்பும் என்பதை நிரல்பெயர்ப்பிக்கு உணர்த்துகிறது. எனவேதான் main( ) செயற்கூறில் அழைப்புக் கூற்று sum=fun(a, b) எனஅமைக்கப்பட்டுள்ளது. Sum என்னும் மாறி return a+b என்னும் கட்டளை மூலமாக int மதிப்பினைப் பெற்றுக் கொள்கிறது.

✓ void தரவினம், அச்செயற்கூறு எந்த மதிப்பையும் திருப்பியனுப்பாது என்பதை நிரல்பெயர்ப்பிக்கு உணர்த்துகிறது. பரந்த சூழலில் நோக்கினால், void தரவினம் எந்த மதிப்பையும் ஏற்காது என்பதை உணர்த்துகிறது.

அடிப்படைத் தரவினங்கள் பல்வேறு பண்புணர்த்திகளை (modifiers) ஏற்கின்றன. இந்தப் பண்புணர்த்திகள் அகநிலையில் தரவுகளை உருவகிப்பதில் ஆழமான விளைவுகளை ஏற்படுத்துகின்றன. signed, unsigned long, short, ஆகியவை அவற்றுள் சில பண்புணர்த்திகள் ஆகும். தரவினங்கள், நினைவகத்தில் அவை எடுத்துக் கொள்ளும் பைட் அளவு, அவற்றின் வரம்பு மதிப்புகள் ஆகியவற்றை அட்டவணை **11.13**ல் காண்க.

Type	Byte	Range
char	1	-128 to 127
unsigned char	1	0 to 255
signed char	1	-128 to 127
int	2	-32768 to 32767
unsigned int, unsigned short int	2	0 to 65535
signed int,short int, signed short int	2	-32768 to 32767
long int,signed long int	4	-2147483648 to 2147483647
unsigned long int	4	0 to 4294967295
float	4	3.4e-38 to 3.4e+38
double	8	1.7e-308 to 1.7e+308
long double	10	3.4e-4932 to 1.1e+4932

அட்டவணை 11.13 தரவினங்களும் அவற்றின் அளவும் வரம்பு மதிப்புகளும்

11.3.4. தருவிக்கப்பட்ட தரவினங்கள் (Derived Data Types)

தருவிக்கப்பட்ட தரவினங்கள் int, float போன்ற உள்ளிணைந்த அடிப்படைத் தரவினங்கள் அல்லது பயனர் வரையறுக்கும் தரவினங்களிலிருந்து உருவாக்கப்படுபவை ஆகும். எடுத்துக்காட்டாக, அணி (Array) வகைத் தரவினத்தின் அறிவிப்பு/மதிப்பிருத்தலைக் காண்க:

int num_array[5] = {1,2,3,4,5};

char dayname[7][3] = {"Sun","Mon","Tue","Wed","Thu","Fri","Sat"};

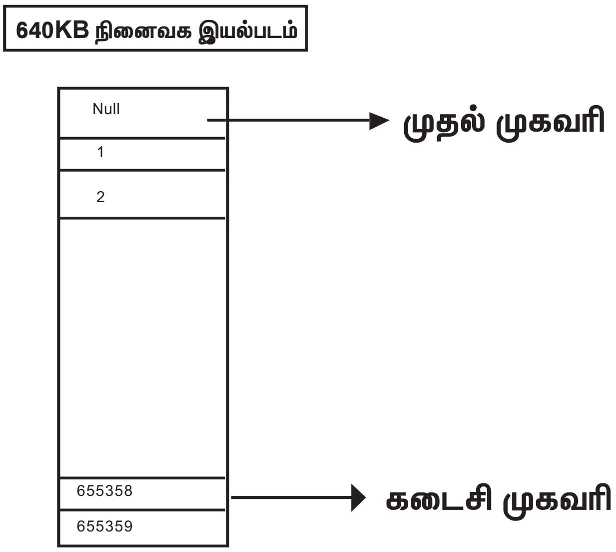
num_array 5 மதிப்புகளைக் கொண்டுள்ளது, அணியிலுள்ள ஒவ்வோர் உறுப்பும் அதன் இருப்பிடச் சுட்டெண் வழியாக அணுகப்படுகிறது. இருப்பிட எண் 0விலிருந்து தொடங்குகிறது. num_array[0] என்னும் உறுப்பில் 1 இருக்கும். num_array[4] என்னும் உறுப்பில் 5 இருக்கும்.

dayname[0], dayname [5], dayname [3][2] ஆகிய உறுப்புகளில் உள்ள மதிப்புகள் என்னென்ன என்பதைக் கூற முடியுமா ?

சுட்டுகள் (Pointers)

சுட்டு என்பது ஒரு நினைவக முகவரியை ஏற்கும் மாறி ஆகும். ஒரு மாறியின் நினைவக இருப்பிடத்தை நேரடியாக அணுகுவதற்கு சுட்டுகள் உதவுகின்றன. கணிப்பொறி நினைவகத்தில் உள்ள ஒவ்வொரு பைட்டும் ஒரு முகவரியைக் கொண்டுள்ளன. முகவரி என்பது ஓர் எண்ணாகும். நமது வீட்டுக் கதவெண்களைப் போல, முகவரி எண் என்பது NULL என்பதில் தொடங்கி 1, 2, 3........என வரிசையாக அமையும்.

எடுத்துக்காட்டாக, 640kb அளவுள்ள நினைவகம், NULL இல் தொடங்கி 6,55,356 வரை முகவரிகளைக் கொண்டிருக்கம். படம் 11.4 காண்க.



படம் 11.4 640 KB நினைவக இயல்படம்

ஒரு நிரலை எந்திர மொழிக்கு மாற்றும்போது, நிரல்பெயர்ப்பி, மாறிகளுக்குரிய நினைவக இருப்பிடங்களை ஒதுக்கீடு செய்கிறது. மாறி சார்ந்துள்ள தரவினத்துக்கு ஏற்ப அதற்கு ஒதுக்கப்படும் நினைவகத்தின் இடஅளவு அமைகிறது.

எடுத்துக்காட்டாக, கீழேயுள்ள அறிவிப்புகளைக் காண்க

char c ; int i; float f;

char c – 1 பைட்
int i – 2 பைட்டுகள்
float f – 4 பைட்டுகள்

> ஒவ்வொரு மாறியும் அதன் முகவரியைக் கொண்டே குறிப்பிடப்படுகிறது. நமது எடுத்துக் காட்டில், c, i, f, ஆகிய மாறிகளின் நினைவக இருப்பிடங்கள் முறையே 1, 2, 4 பைட்டுகளைக் கொண்டிருக்கும். இவற்றின் முகவரிகள் பதினறும எண் முறையில் குறிக்கப்படுகின்றன.

சுட்டுத் தரவினத்தைக் கையாள்பவர், **முகவரி சுட்டல் (&), மதிப்பு சுட்டல் (*)** ஆகிய செயற்குறிகளைப் பற்றி அறிந்துகொள்ள வேண்டும்.

‘ &’ செயற்குறி:

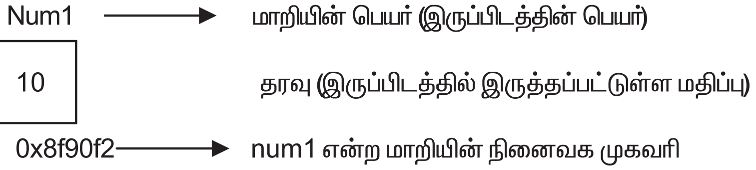
int num1=10;

என்ற கட்டளையைத் தந்தால், சி++ நிரல்பெயர்ப்பி (C++ Complier) கீழ்காணும் செயல்பாடுகளை செய்து முடிக்கின்றது.

1. நினைவகத்தில் ஓர் int மதிப்பை இருத்திவைக்க இடம் ஒதுக்கீடு செய்கிறது.

2. அந்த நினைவக இடத்துக்கு num 1 என்ற பெயரைச் சூட்டுகிறது.

3. நினைவகத்தில் அந்த இடத்தில் 10 என்னும் மதிப்பை இருத்துகிறது,




```
// Program – 11.1
// to demonstrate use of & operator
#include<iostream.h>
#include<conio.h>
void main()
{        clrscr();
          int i = 10;

cout << “\n Address of the variable... “ <<&i;
cout << “\nValue stored in the variable ..” << i;

          getch();
}
```

கீழேயுள்ள கட்டளைத் தொகுதியைக் காண்க.

```
int *x, num 1;
num1 = 10;
x=&num1;
cout<<*x;
```

குறிப்பு: நட்சத்திரக் குறி (*)

1. ஒரு சுட்டு இன மாறியை அறிவிக்கப் பயன்படுத்தப்பட்டுள்ளது.
2. நினைவகத்தில் இருத்தப்பட்டுள்ள மதிப்பைத் திரையில் காட்டப் பயன்பட்டுள்ளது (மதிப்பு சுட்டல் செயற்குறி).
3. இது ஓர் ஒருமச் செயற்குறி (Unary Operator) ஆகும்.

11.4 மாறிகள் (Variables)

குறிப்பிட்ட வரம்பெல்லைக்குள் ஒரு மதிப்பினை ஏற்கின்ற ஒரு தரவுப் புலத்தின் பெயரே மாறி எனப்படுகிறது. எடுத்துக்காட்டாக, கீழேயுள்ள கூற்றுகளைக் காண்க.

```
int num;

num = 5;
```

int num ; என்னும் கூற்று ''n என்பது int இனத்தைச் சேர்ந்த மாறி'' என்ற பொருளை உணர்த்துகிறது.

```
int num;

num = 5;
```

என்னும் மதிப்பிருத்து கூற்று, ''5 என்னும் மதிப்பு num என்னும் மாறியில் இருத்தப்படுகிறது'' என்ற பொருளை உணர்த்துகிறது.

> தரவுகள் இருத்தப்படும் நினைவக இருப்பிடங்களுக்குப் பயனர் சூட்டும் பெயர்களே **மாறிகள்** ஆகும்.

மாறியின் பெயர்கள் எழுத்துகள், எண்கள், அடிக்கீறு (underscore)ஆகியவற்றைக் கொண்டிருக்கலாம். பெயர்கள் எழுத்தில் அல்லது அடிக்கீறில் தொடங்க வேண்டும். எனினும், அடிக்கீறில் தொடங்கும் பெயர்கள் அகநிலை முறைமை மாறிகளுக்காக (internal system variables) ஒதுக்கப்பட்டுள்ளன என்பதை அறிக. பெயர்கள் எழுத்து வடிவ உணர்வு (case sensitive) உள்ளவை. அதாவது, சிறிய பெரிய எழுத்தில் அமையும் பெயர்கள் வேறுபடுத்தி அறியப்படுகின்றன. அட்டவணை 11.14–ஐ நிறைவு செய்க.

மாறி	சரியானதா/ பிழையானதா ?	பிழையானது என்பதற்கான காரணம்
A_b	சரியானது	
1a_b	பிழையானது	மாறியின் பெயர் எழுத்து அல்லது அடிக்கீறில்தான் தொடங்க வேண்டும்.
_Test		
Balance$		
#One		
Include		

அட்டவணை 11.14 மாறிப் பெயர்களின் ஏற்புத் தகைமை

11.4.1 மாறிகளின் அறிவிப்பு

மதிப்புகளை இருத்திக்கொள்ள மாறிகளுக்கு நினைவகம் ஒதுக்கப்படுகிறது. மாறி சார்ந்துள்ள தரவினத்தின் அடிப்படையில் நிரல்பெயர்ப்பி நினைவகத்தை ஒதுக்குகிறது. எனவே, மாறிகளைப் பயன்படுத்துவதற்கு முன்பாக அவற்றைத் தரவினம் குறிப்பிட்டு அறிவிக்க வேண்டும்.

எடுத்துக்காட்டு

```
int q;
float f1,f2;
char name[10],choice;
```

அறிவிப்புக் கூற்றின் அமைப்பு :

int side, float hypotenuse , area ;

என்னும் அறிவிப்பை எடுத்துக் கொள்வோம். இது தவறான அறிவிப்பாகும். காரணம், நிரல்பெயர்ப்பி இக்கூற்றினை இவ்வாறு புரிந்துகொள்ளும்.

- side, float, hypotenuse, area ஆகியவை int இன மாறிகளாகக் கருதப்படும். எனவே, "float" என்னும் சொல்லுக்குப் பிறகு காற்புள்ளி இடம்பெற வேண்டும்" என்னும் பிழைசுட்டும் செய்தியைத் தரும்.
- side என்னும் int இன மாறியையும், hypotenuse area என்னும் float இன மாறிகளையும் அறிவிப்பதே நமது நோக்கம். எனவே மேற்கண்ட அறிவிப்புக் கூற்றினை இவ்வாறு அமைக்க வேண்டும்.

int side ; **float** hypotenuse , area ;	***int*** side ; ***float*** hypotenuse,area ;

> ✓ ஒரே இனத்தைச் சார்ந்த ஒன்றுக்கு மேற்பட்ட மாறிகளை ஒரே அறிவிப்புக் கூற்றில் குறிப்பிட முடியும். ஆனால் காற்புள்ளியால் பிரிக்கப்பட்டிருக்க வேண்டும்.

தரவினங்களை அறிவிப்பதற்கென char, int, double, float, void, signed, long, unsigned ஆகிய ஒன்பது சொற்கள் பயன்படுத்தப்படுகின்றன.

long, short, signed, unsigned ஆகியவை, உள்ளிணைக்கப்பட்ட தரவு இனங்களின் பண்புகளை மாற்றியமைக்கப் பயன்படும் பண்புணர்த்திகள் (modifier) அல்லது தகுதியாக்கிகள்(qualifiers) எனப்படுகின்றன. ஆனால் void இனத்துக்கு இவற்றைப் பயன்படுத்த முடியாது.

அகநிலையில் **15** என்னும் int மதிப்பு **00000000 00001111** என்று கையாளப்படுகிறது. int மதிப்புகள் இரும எண்முறையில் **16** பிட் வடிவமைப்பில் இருத்தி வைக்கப்படுகின்றன. வலப்புறம் தொடங்கி ஒவ்வொரு பிட்டாக இடப்புறம் அமைந்து மொத்தம் **15** பிட்டுகள் மதிப்பை இருத்தப் பயன்படுத்தப்படுகின்றன. எனவே ஓர் int மாறியில் இருத்தப்படும் பெரும மதிப்பு +**32767.** சிறும மதிப்பு **–32768.** (2^{15} = **32768.** நேர்ம மதிப்பு **0** முதல் + **32767** வரை. எதிர்ம மதிப்பு **–1** முதல் **–32768** வரை) **16–**வது பிட், உச்ச மதிப்பு பிட் (Most Significant Bit) அல்லது குறி பிட் (Sign Bit) எனப்படுகிறது. இது, அவ்வெண் நேர்மம் (positive) அல்லது எதிர்மம் (Negative) என்பதைக் குறிக்கிறது. **16–**வது பிட் **1** எனில் அவ்வெண் எதிர்ம எண், **0** எனில் நேர்ம எண் ஆகும்.

ஓர் int இன மாறியின் அறிவிப்பில் unsigned என்னும் பண்புணர்த்தியைப் பயன்படுத்தினால் அம்மாறியில் நேர்ம மதிப்புகளை மட்டுமே இருத்த முடியும். குறி பிட்டும் தரவு மதிப்புகாகவே பயன்படுத்திக்கொள்ளப்படும். எனவே, மதிப்பின் வரம்பெல்லை 2^{16} வரை இருக்க முடியும். அதாவது **0** முதல் **65535** வரை இருத்தி வைக்கலாம்.

✓ பண்புணர்த்தி **(modifier)** ஓர் அடிப்படைத் தரவினத்தின் பண்பினை மாற்றியமைத்துப் புதிய தரவினத்தை உருவாக்குகிறது.

பண்புணர்த்திகளின் விளைவுகள்

- **unsigned** என்னும் பண்புணர்த்தி, நேர்மம்/ எதிர்மம் என்பதைக் குறிக்கும் குறி பிட்டையும் மதிப்புணர்த்தப் பயன்படுத்திக் கொள்வதால் மாறி ஏற்கும் மதிப்பின் வரம்பெல்லை மாற்றியமைக்கப்படுகிறது.
- **long** என்னும் பண்புணர்த்தி குறிப்பிட்ட தரவினத்தின் பைட்டுகளை அதிகரித்து, மதிப்பின் வரம்பெல்லையை நீட்டிக்கிறது.

அடிப்படைத் தரவினத்தில் மாறிகள் அறிவிக்கப்படும்போது, முன் னொட்டாக பண்புணர்த்திகள் இடம்பெற வேண்டும். எடுத்துக்காட்டாக,

unsigned int registration_number;

unsigned long int index;

signed char c

✓ மாறிகளை அறிவிக்கும்போது, பண்புணர்திகளைத் தரவினப் பெயருக்கு முன்னொட்டாகத் தர வேண்டும்.

const என்னும் தகுதியாக்கி (qualifier) நிரலின் இயக்க நேரத்தில் ஒரு மாறியின் மதிப்பு மாற்றப்பட முடியாது என்பதை உணர்த்துகிறது. const மாறியின் மதிப்பை மாற்ற முயன்றால் நிரல்பெயர்ப்பி பிழை சுட்டும் செய்தியைத் தரும். const தகுதியாக்கி, பிற பண்புணர்த்திகளைப் போன்றே மாறிகள் அறிவிக்கப்படும்போது தரவினப் பெயருக்கு முன்னொட்டாக இடம் பெறுகிறது.

எடுத்துக்காட்டு.

```
const float pi = 3.14;
```

தகுதியாக்கிகளும் பண்புணர்த்திகளும் பயன்படுத்தும்போது பிட் எண்ணிக்கையும் மதிப்பின் வரம்பெல்லையும் மாற்றியமைக்கப்படும் தரவினங்களை அட்டவணை **11.15–**ல் காண்க.

Data Types		
Type	Length	Range
unsigned char	8 bits	0 to 255
char	8 bits	-128 to 127
enum	16 bits	-32,768 to 32,767
unsigned int	16 bits	0 to 65,535
short int	16 bits	-32,768 to 32,767
int	16 bits	-32,768 to 32,767
unsigned long	32 bits	0 to 4,294,967,295
long	32 bits	-2,147,483,648 to 2,147,483,647
float	32 bits	3.4×10^{-38} to 3.4×10^{38}
double	64 bits	1.7×10^{-308} to 1.7×10^{308}
long double	80 bits	3.4×10^{-4932} to 1.1×10^{4932}

அட்டவணை 11.15 பண்புணர்த்திகளுடன் தரவினங்கள்

சுட்டு மாறிகளை அறிவித்தல்

int * iptr;

→ **சுட்டு மாறியின் பெயர்**

→ **இந்த மாறி, சுட்டு இனத்தைச் சேர்ந்தது என்பதை நிரல்பெயர்ப்பிக்கு உணர்த்துகிறது (மாறி ஒரு முகவரியை ஏற்கும்)**

→ **இந்தச் சுட்டு ஓர் int இனத் தரவைச் சுட்டுகிறது என்பதைக் குறிக்கிறது.**

int * iptr என்னும் அறிவிப்புக் கூற்றுக்கு, iptr என்பது int இனத்தைச் சுட்டும் ஒரு சுட்டு மாறி (pointer variable) என்று பொருள்கொள்ள வேண்டும். int இன மதிப்புகள் இருத்திவைக்கப்பட்டுள்ள ஒரு முகவரியை மட்டு iptr மாறி ஏற்கும்.

சுட்டு மாறியின் அறிவிப்புகளுக்கான எடுத்துக்காட்டுகள்.

char * cptr float * fptr	char இனத்தைச் சுட்டும் ஒரு சுட்டின் அறிவிப்பு float இனத்துக்கான ஒரு சுட்டு
void *v ptr	எந்தத் தரவினத்தையும் சுட்டுகின்ற ஒரு சுட்டு. ஒரு பொதுநிலைச் சுட்டு இவ்வாறு அறிவிக்கப்படுகிறது.
const int * ptr	ஒரு மாறிலி int மதிப்புக்கான சுட்டு ptr . (ptr சுட்டும் நினைவக முகவரியிலுள்ள int மதிப்பை மாற்றியமைக்க முடியாது)

char * const cp	cp என்பது ஒரு மாறிலிச் சுட்டு. cp - ல் இருத்தப்பட்டுள்ள முகவரியை மாற்றியமைக்க முடியாது.

அட்டவணை 11.16 சுட்டு மாறிகளுக்கு எடுத்துக்காட்டுகள்

11.4.2 மாறிகளைத் தொடங்கிவைத்தல் (Initialization of variables)

மாறிகளை அறிவிக்கும்போதே அதில் குறிப்பிட்ட மதிப்பு இருத்தப்படுவதைத் **தொடக்க மதிப்பிருத்தல்** என்கிறோம். மாறிகளில் தொடக்க மதிப்பிருத்தலையே **தொடங்கிவைத்தல்** (Initialization)என்கிறோம்.

```
int num = 10;
int fun(5);
```

முதலாவது கூற்றில் num என்னும் மாறியில் 10 என்னும் தொடக்க மதிப்பிருத்தப்பட்டுள்ளது. இரண்டாவது கூற்றில் ஓர் ஆக்கி (constructor) மூலம் 5 என்னும் மதிப்பு fun என்னும் மாறியில் இருத்தப்பட்டுள்ளது.

உள்ளுறை இனமாற்றங்கள் (Implicit conversions):

ஒரு கோவையில் இடம்பெறும் தரவுகளின் இனம் நிரல்பெயர்ப்பியால் மாற்றப்படுவது உள்ளுறை இனமாற்றம் எனப்படுகிறது. எடுத்துக்காட்டாக, கீழேயுள்ள கட்டளைத் தொகுதியைக் காண்க.

```
float f = 7.6;
int x = f;
```

இப்போது, x என்னும் மாறியில் 7 என்னும் மதிப்பு இருக்கும். float இனம் int இனமாக மாற்றப்பட்டுவிட்டது. நிரல்பெயர்ப்பி இந்த இனமாற்றத்தைத் தானாகவே செய்கிறது.

உள்ளுறை இனமாற்றத்துக்கான விதிமுறைகள் :

ஒரு செயற்குறியால் இணைக்கப்பட்டுள்ள இரண்டு செயலேற்பிகளையும் ஒரு செயற்குறியையும் கொண்ட ஒரு கோவையை எடுத்துக்கொள்ளுங்கள். அதில் கீழேயுள்ளவாறு இனமாற்றங்கள் நிகழும்.

1. ஒரு செயலேற்பி long double இனம் எனில் இன்னொரு செயலேற்பியின் மதிப்பும் long double இனமாக மாற்றப்படும்.
2. ஒரு செயலேற்பி double இனம் எனில் இன்னொரு மதிப்பும் double இனமாக மாற்றப்படும்.
3. ஒரு செயலேற்பி float இனம் எனில் இன்னொரு மதிப்பும் float இனமாக மாற்றப்படும்.
4. ஒரு செயலேற்பி unsigned long int இனம் எனில் இன்னொரு மதிப்பும் unsigned long int இனமாக மாற்றப்படும்.
5. ஒரு செயலேற்பி long int இனம் எனில் இன்னொரு மதிப்பும் long int இனமாக மாற்றப்படும்.
6. ஒரு செயலேற்பி unsigned int இனம் எனில் இன்னொரு மதிப்பும் unsigned int இனமாக மாற்றப்படும்.

```
// demonstrating implicit type conversions
// Program – 11.2
# include <iostream.h>
# include <conio.h>
# include <iomanip.h>
void main()
{
clrscr();
int i; float f;
double d;
long double ld;
unsigned int ui;
unsigned long int uli;
i = -5;
f = 2;
d = 3;
ld = 3;
ui = 6;
uli = 4;
cout <<"\nSizeof long double.."<<sizeof(ld*d)<<'\t'<<ld*d;
cout << "\nSizeof double..." << sizeof(d*f)<<'\t'<<d*f;
cout << "\nSizeof float..." << sizeof(f * i) <<'\t' << f*i;
cout << "\nSizeof unsigned long int ..." << sizeof(uli* f)<<'\t'<< uli * f;
cout << "\nSizeof unsigned int..." << sizeof(ui * i) <<'\t'<< ui * i;
getch();
}
```

குறிப்பு: sizeof என்பது ஒரு செயற்குறி. தரப்படும் ஒரு கோவை அல்லது தரவினத்தின் அளவினை (அத்தரவின் மதிப்பை நினைவகத்தில் இருத்தத் தேவைப்படும் இடத்தை) பைட்டுகளில் விடையாகத் தரும்.

Output displayed by the above program:		
Sizeof long double	**...10**	**9**
Sizeof double	**...8**	**6**
Sizeof float	**...4**	**-10**
Sizeof unsigned long int	**...4**	**8**
Sizeof unsigned int	**...2**	**65506**

program 11.2 –ன் அடிப்படையில் அட்டவணை 11.17 –ஐ நிறைவு செய்க. காரணம் என்னும் நெடுக்கையில் முதல் மதிப்புக்குத் தரப்பட்டுள்ள விளக்கத்தைப் போல விடைகளை எழுதுக.

வரிசை எண்	விடையின் பைட் அளவு	கோவை	காரணம்
1.	10	ld*d	id என்னும் மாறி long double என்பதால் கோவையிலிருந்து பெறப்படும் மதிப்பும் long double இனமாக இருக்கும்.
2.	8	d*f	________________________ ________________________ கோவையிலிருந்து பெறப்படும் மதிப்பும் double இனமாக இருக்கும்.
3.	4	f*l	
4.	4	uli * f	
5.	2	ui * i	

அட்டவணை 11.17 program **11.2–ன் அடிப்படையிலான பயிற்சி**

சுட்டு மாறிகளைத் தொடங்கிவைத்தல்

சுட்டுமாறிகள் பிற மாறிகளின் முகவரியை இருத்திவைக்கின்றன. ஆனால், ஒரு சுட்டு மாறி எந்த இனத் தரவினைச் சுட்டுவதற்காக உருவாக்கப்பட்டதோ, அதே இன மாறியின் முகவரியையே ஏற்றுக் கொள்ளும். எடுத்துக்காட்டாக,

```
int *iptr, num1;
num1 = 10;
iptr = &num1; //சுட்டு மாறியில் தொடக்க மதிப்பிருத்தல்.
```

கீழே காணும் தொடக்க மதிப்பிருத்தல் பிழையானதாகும்

```
int *iptr;
float num1 = 10.5;
iptr = &num1    // int இனத்தைச் சுட்டும் iptr என்னும் சுட்டு மாறியில் float இன மாறியின் முகவரியை இருத்த முயல்வதால் பிழைசுட்டும் செய்தி கிடைக்கும்.
```

சுட்டு மாறிகள் அவை சுட்டும் மாறியின் தரவினத்தோடு ஒத்திருப்பது கட்டாயமாகும்.

இனமாற்றம் (Type cast) **:**

ஒரு மாறியில் இருத்திவைக்கப்பட்டுள்ள மதிப்பின் தரவினத்தை மாற்றிப் பயன்படுத்தும் செயலாக்கம் **இனமாற்றம்** என்று அறியப்படுகிறது (float)7 என்னும் கூற்று 7 என்னும் முழுஎண் மதிப்பை float இனமாக மாற்றுகிறது. ஒரு மாறி அல்லது மதிப்பின் முன்னொட்டாகத் தேவையான தரவினத்தைக் குறிப்பிடுவதன் மூலம் இனமாற்றம் எய்தப்படுகிறது.

கட்டளை அமைப்பு :

(< தரவு இனம்>) <மாறி/மதிப்பு>

அல்லது

< தரவு இனம்> (<மாறி/மதிப்பு>)

இனமாற்றம் என்பது, உள்ளிணைந்த அடிப்படைத் தரவினங்களுக்கு மட்டுமே உரியது.

x = 8% 7.7 ;

என்னும் கட்டளை அமைப்பு நிரல்பெயர்ப்பின் (compilation) போது ஒரு பிழைசுட்டும் செய்தியைத் தரும். காரணம், % என்னும் வகுமீதி செயற்குறி int தரவின மதிப்புகளின் மேல்தான் செயல்படும். மேற்கண்ட பிழையான கட்டளையை,

x = 8% (int) 7.7;

எனத் திருத்தி அமைக்கலாம். float மாறிலி 7.7 இனமாற்றத்தின் மூலம் int மாறிலியாக மாற்றப்பட்டுள்ளது. அட்டவணை 11.18ஐ நிறைவு செய்க.

int x; x =7 / 3;	x–ல் இருத்தி வைக்கப்பட்டுள்ள மதிப்பு யாது ?
float x; x = 7 / 3;	x–ல் இருத்தி வைக்கப்பட்டுள்ள மதிப்பு யாது ?
float x; x = 7.0 / 3.0;	x–ல் இருத்தி வைக்கப்பட்டுள்ள மதிப்பு யாது ?
float x; x = (float) 7 / 3;	x–ல் இருத்தி வைக்கப்பட்டுள்ள மதிப்பு யாது ?

float x;
int a = 7 , b = 3;
x = a/b;
	x–ல் இருத்தி வைக்கப்பட்டுள்ள மதிப்பு யாது ?

float x;
int a = 7 , b = 3;
x = a/ (float) b;
	x–ல் இருத்தி வைக்கப்பட்டுள்ள மதிப்பு யாது ?

அட்டவணை 11.18 x–ன் மதிப்பைக் கண்டறிக.

பயிற்சி வினாக்கள்

1. கீழ்காணும் கோவைகளில் மதிப்பிடல் வரிசையைத் தீர்மானிக்கவும்.

   i. a + pow(b,c) * 2

   ii. a || b && c

   iii. a<b && c || d > a

   iv. (c>=50)||(!flag)&&(b+5 == 70)

   v. (a+b)/(a-b)

   vi. (b*b) – 4 * a * c

2. கீழ்காணும் நிரல்களில் பிழைகளைக் சுட்டிக் காட்டுக.

a.
```
# include <iostream.h>
void main()
{
  float f = 10.0;
  x = 50;
  cout << x << f;
}
```

b.
```
# include <iostream.h>
void main()
{
  float f = 10.0;
  x = 50;
  cout << x << f;
}
```

c.
```
# include <iostream.h>
void main()
{
  int x,y,k,I;
  x =  y + k------I;
  cout << x;
}
```

3. இந்த நிரல்களின் வெளியீடு என்னவாக இருக்கும் ?

a.
```
# include <iostream.h>
# include <conio.h>

void main()
{
  int i=20;
  cout << i << i++ << ++i;
  getch();
}
```

b.
```
# include <iostream.h>
# include <conio.h>
void main()
{
  clrscr();
  int i = 1, a= 3;
  i = a++;
  cout << i;
  getch();
}
```

c.
```
# include <iostream.h>
# include <conio.h>

void main()
{
  clrscr();
  int i = 3,x;
  x = i ? i++ : ++i;
  cout << x;
  getch();
}
```

d.
```
# include <iostream.h>
# include <conio.h>
void main()
{
  int z,x = 3, y = 2;
  z = --x + y++;
  cout << z;
  getch();
}
```

e.
```
# include <iostream.h>
# include <conio.h>

void main()
{
  clrscr();
  char ch = 'a';
  ch = (ch == 'b') ? ch :' b';
  cout << ch;
  getch();
}
```

4. கீழ்காணும் சி++ மொழிக் கோவைகளை மதிப்பிடுக.

a = 5, b =3, d =1.5, int c, float f எனக் கொள்க.

a. f = a+b/a

b. c = d * a+b

c. x = a++ * d + a;

d. y = a – b++ * —b;

e. (x >= y) ||(!(z==y) && (z < x)) எனில்

✓ x = 10, y = 5, z = 11 (அனைத்தும் முழு எண்கள்)

✓ x = 10, y = 10, z = 10

✓ x = 9, y = 10, z = 2

5. கீழ்க்காணும் நிபந்தனைகளை, நிபந்தனைச் செயற்குறியைப் பயன்படுத்தி சி++ கோவையாக எழுதிக் காட்டுக.

✓ x = 30 எனில் f = 0.5, இல்லையேல் f = 5

✓ x > = 60 எனில் fn = 0.9, இல்லையேல் .7

6. சுட்டு மாறிகள் என்றால் என்ன ?

7. ஒரு குறியுருத் தரவின மாறியின் முகவரியை இருத்தி வைக்கும் name என்னும் சுட்டு மாறியை அறிவிக்கும் கூற்றினை எழுதிக் காட்டுக.

பாடம் 12

அடிப்படை கூற்றுகள்

(Basic Statements)

சி++ மொழியிலுள்ள அடிப்படைக் கூற்றுகள் (Basic Statements) வில்லைகளைப் (tokens) பயன்படுத்தி உருவாக்கப்படுகின்றன. சி++ மொழிக் கூற்றுக்களைப் பல்வேறு வகையாகப் பிரிக்கலாம்.

- உள்ளீடு/வெளியீடு (Input/Output)
- அறிவிப்பு (Declaration)
- மதிப்பிருத்தல் (Assignment)
- கட்டுப்பாட்டுக் கட்டமைப்புகள் (Control Structures)
- செயற்கூறு அழைப்பு (Function Call)
- பொருளின் செய்தி (Object Message)
- திருப்பியனுப்பல் (Return)

12.1 உள்ளீடு/வெளியீட்டுக் கூற்றுகள்

தரவுகளைப் பெற்றுக் கொண்டு அவற்றைச் செயற்படுத்தித் தகவலாக வெளியிடுவதே எந்தவொரு கணிப்பொறி நிரலின் இன்றியமையாத செயல்பாடாகும். மாறிகளில் தரவுகளை இருத்தி வைக்க இரண்டு வழி முறைகள் உள்ளன. அவற்றுள் ஒன்று, மதிப்பிருத்து கூற்று மூலமாகச் செய்யப்படுவது. இதனை முந்தைய பாடத்தில் ஏற்கெனவே படித்துள்ளோம். இன்னொரு வழிமுறை, நிரலின் இயக்க நேரத்தில் உள்ளீடாகப் பெற்றுக் கொள்வது, நிரல் இயங்கிக் கொண்டிருக்கும் போது, விசைப்பலகை வழியாகத் தரவுகளை உள்ளீடாகப் பெற **cin-** என்னும் **பொருள்** (சி – இன் என உச்சரிக்க வேண்டும்) பயன்படுகிறது. cin என்பது சி++ மொழியில் முன்வரையறுக்கப்பட்டுள்ள ஒரு **பொருள்** (Object). இது **அடிப்படை உள்ளீட்டுத் தாரையை** (Standard input Stream) உருவகிக்கிறது. உள்ளீட்டுத் தாரை என்பது, அடிப்படை உள்ளீட்டுச் சாதனமான விசைப் பலகையிலிருந்து பெறப்படும் தரவுகளை குறிக்கிறது, cin வேறுவகை உள்ளீட்டு மூலங்களிலிருந்தும் தரவுகளை படிக்கும், இதைப் பற்றிப் பிறகு பார்ப்போம். Cin என்னும் பொருளைப் பற்றி அறிவிப்புகள் istream.h என்னும் தலைப்புக் கோப்பில் (Header file) தரப்பட்டுள்ளன. அடிப்படையான உள்ளீட்டு/வெளியீட்டுச் செயல்பாடுகள் istream.h, ostream.h ஆகிய தலைப்புக் கோப்புகளில் வரையறுக்கப்பட்டுள்ள சில அறிவிப்புகளின் மூலம் நிறைவேற்றப்படுகின்றன. istream, ostream ஆகிய இரண்டின் பண்புக் கூறுகளையும் iostream.h என்னும் கோப்பு தன்னகத்தே கொண்டுள்ளது.

- தலைப்புக் கோப்பு, முன்வரையறுக்கப்பட்ட செயல்கூறுகளின் அனைத்து அடிப்படை அறிவிப்புகளையும் வரையறைகளையும் கொண்டுள்ளது.
- முன்–செயலி நெறியுறுத்தத்தைப் பயன்படுத்தி, ஒரு தலைப்புக் கோப்பினை நிரலில் இணைத்துக் கொள்ள முடியும்.

- முன்–செயலி நெறியுறுத்தம் # என்ற குறியுடன் தொடங்கும். இது, தேவையான பணியைச் செய்யுமாறு நிரல்பெயர்ப்பிக்கு உணர்த்துகிறது
- வழக்கமாய் ஒரு முன்செயலி நெறியுறுத்தம் # include<iostream.h> என்று அமையும். இக்கூற்று, iostream.h என்னும் தலைப்புக் கோப்பினை நிரலில் சேர்த்துக் கொள்ளுமாறு நிரல் பெயர்ப்பிக்கு உணர்த்துகிறது. cin/cout ஆகிய பொருள்களைப் பயன்படுத்திக் கொள்ள வேண்டுமெனில், iostream.h என்னும் கோப்பினை நிரலில் இணைத்துக் கொள்ள வேண்டும்.
- வேறுசில தலைப்புக் கோப்புகள்: iomanip.h, stdio.h, ctype.h, math.h, fstream.h மற்றும் பிற

>> என்னும் செயற்குறி **தரவு ஈர்ப்பு** (extraction) அல்லது **தரவு பெறும்** (get from) செயற்குறி ஆகும். இச்செயற்குறி, தனக்கு இடப்புறமுள்ள தாரைப் பொருளிலிருந்து (stream object) தரவு மதிப்புகளை எடுத்து, வலப்புறமுள்ள மாறியில் இருத்தி வைக்கும். எடுத்துக்காட்டாக, கீழேயுள்ள கட்டளைகளைக் காண்க.

```
float temperature;

cin >> temperature;
```

தரவு ஈர்ப்புச் செயற்குறி (>>) உள்ளீட்டுத் தாரைப் பொருளான cin லிருந்து தரவினை ஈர்த்தெடுத்து, வலப்புறமுள்ள temperature என்னும் மாறியில் இருத்தி வைக்கிறது. தரவு ஈர்ப்புச் செயற்குறியை அடுத்தடுத்து அமைத்து, ஒன்றுக்கு மேற்பட்ட தரவு மதிப்புகளை உள்ளீட்டுத் தாரையிலிருந்து பெற்று, உரிய மாறிகளில் இருத்தி வைக்க முடியும். எடுத்துக்காட்டாக, temperature, humidity ஆகிய இரு மாறிகளுக்குத் தரவுகளை உள்ளீடாகப் பெறுவதற்கு,

```
cin >> temperature >> humidity;
```

எனக் கட்டளை அமைக்க வேண்டும்.

cout என்பது (சி–அவுட் என உச்சரிக்க வேண்டும்) அடிப்படை வெளியீட்டுத் தாரைக்கென முன்வரையறுக்கப்பட்டுள்ள பொருள் (Object) ஆகும். அடிப்படை வெளியீட்டுத் தாரை திரைக் காட்சியாகப் பாய்கிறது. வேறு பல வெளியீட்டுச் சாதனங்களுக்கும் இத்தாரையைத் திசைதிருப்ப (redirect) முடியும். << என்னும் செயற்குறி **தரவு விடுப்பு** (insertion) அல்லது **தரவுதரும்** (put to) செயற்குறி ஆகும். இச்செயற்குறி தனக்கு வலப்பக்கம் உள்ள ஒரு மாறியின் மதிப்பை இடப்பக்கமுள்ள பொருளில் இருத்தி வைக்கும். எடுத்துக்காட்டாக கீழேயுள்ள கட்டளைகளை நோக்குங்கள் :

```
int marks = 85;

cout << marks;

cout << "\n Marks  obtained is : " << marks;
```

marks என்னும் மாறியில் சேமிக்கப்பட்டுள்ள மதிப்பு cout பொருளுக்கு அனுப்பி வைக்கப்படும். marks ன் மதிப்பு திரையில் காட்டப்படும்.

```
cout << "\n Marks obtained is : " << marks;
```

என்னும் இரண்டாவது கட்டளை, செய்தியையும் மாறியின் மதிப்பையும் திரைக்கு அனுப்பிவைக்கும். இவ்வாறு, வெளியீட்டுச் செயற்குறியை அடுத்தடுத்து அமைத்து, ஒன்றுக்கு மேற்பட்ட விவரங்களை ஒரே கட்டளையில் வெளியிட முடியும்.

இன்னும் சில எடுத்துக்காட்டுகள்

```
cout << "\n The sum of the variables a,b .." << a+b;
cout << a+b << '\t' << a-b << '\t' << a/b;
cout << "\n The difference of numbers …." << a-b << "\n
The sum of two numbers …. " << a+b;
```

12.2 முதல் சி++ நிரல் – சி++ நிரலின் கட்டமைப்பு

```
// My first program – Program 12.1
# include <iostream.h>//preprocessor directive
# include <conio.h>
float fact = 1; // declaration of variables
int term;
int main()      // function header
{
clrscr(); // predefined function
cout << "\n This program computes factorial of a number";
cout << '\n' << "Enter a number ...";
cin >> term;
for(int x = 2; x <= term;fact *= x,x++);// looping statement
cout << "\nThe factorail of the given number .."
<< term << "  is .." << fact;
return 0;
}
```

மேலேயுள்ள நிரலை நோக்குக. ஒரு சி++ நிரல் மூன்று முக்கிய பிரிவுகளைக் கொண்டுள்ளது.

- சேர்த்துக் கொள்ளும் கோப்புகள்
- மாறிகள், பயனர் வரையறுக்கும் செயற்கூறுகளின் அறிவிப்பு
- main() செயற்கூறு.

பிழையேதுமின்றி நிரல்பெயர்ப்புச் செய்தபின், நிரலை இயக்கும் போது main() செயற்கூறு தானாகவே செயல்படுத்தப்படும். இந்தச் செயற்கூறில்தான், நிரலின் பிற கூறுகளையும், பிற செயலாக்கு கட்டளைகளையும் அழைக்கின்ற கூற்றுகள் இடம்பெற வேண்டும்.

12.3 அழைப்புக் கூற்றுகள் (Declaration Statements)

ஒரு நிரலிலுள்ள மாறிகள் பயன்படுத்தப்படுவதற்கு முன்பாக அறிவிக்கப்பட்டு வரையறுக்கப்பட வேண்டும்.

int *iptr; // int இனத்தைச் சுட்டும் ஒரு சுட்டினை அறிவிக்கிறது.
iptr = new int;//int இனத் தரவை இருத்திவைக்க நினைவக இடம் ஒதுக்கப்படுகிறது – எனவே சுட்டு மாறி வரையறுக்கப்பட்டு விட்டது.
*iptr = 5; //5 எனினும் மதிப்பு இருத்தப்படுகிறது. நினைவகம் ஒதுக்கப்பட்ட பிறகே இது நிகழ்கிறது.

ஒரு மாறியின் அறிவிப்பு, மாறியின் பெயரையும் அது சார்ந்த தரவு இனத்தையும் அறிமுகப்படுத்துகிறது. எடுத்துக்காட்டாக, int*iptr, என்னும் அறிவிப்பை எடுத்துக் கொள்ளுங்கள். 'iptr' என்பது int இனத்தைச் சுட்டும் ஒரு சுட்டு மாறி' என இக்கூற்றினைப் புரிந்துக் கொள்ள வேண்டும். அனைத்துச் சுட்டு மாறிகளும், தரவு மதிப்பை இருத்திவைப்பதற்குரிய நினைவகம் ஒதுக்கப்படும் போது தான் வரையறுக்கப்படுகின்றன.

பயனர் வரையறுக்கும் தரவினங்கள், குறிப்பெயர்கள், செயற்கூறு தலைப்புகள் சுட்டு மாறிகள் மற்றும் இவை போன்றவற்றை அறிவிப்பதற்கு அறிவிப்புச் கூற்றுகள் பயன்படுகின்றன. பாடப்பிரிவு **11.3**ல் பயனர் வரையறுக்கும் தரவினங்களைப் பற்றிப் படித்ததை நினைவு கூர்க.

ஒரு மாறியை அறிவிக்கும்போதே, நினைவகம் ஒதுக்கப்படுகிறது எனில், அந்த அறிவிப்பை வரையறை (definition) என்றே சொல்லலாம். எடுத்துக் காட்டாக.

int num;

என்னும் அறிவிப்புக் கூற்றினை நோக்குக. இந்தக் கூற்று, வரையறைக் கூற்று என்றே கூறப்படுகிறது. காரணம், தரவினை இருத்திவைக்க நினைவகம் ஒதுக்கப்பட்டு விடுகிறது. கீழேயுள்ள கட்டளைத் தொகுதியை நோக்குங்கள்.

int num// ஓர் int மாறியின் அறிவிப்பும் வரையறையும்

num = 5// மதிப்பு **5** இருத்தப்படுகிறது

நினைவக ஒதுக்கீட்டுக்கென வெளிப்படையான கோரிக்கை எதுவும் இல்லை என்பதைக் கவனித்தீர்களா **?** காரணம் char, int, float போன்ற அடிப்படைத் தரவினங்களில் ஒரு மாறியை அறிவிக்கும்போதே நினைவகம் ஒதுக்கப்பட்டு விடுகிறது.

- ✓ அறிவிப்புக் கூற்று ஒரு மாறியின் பெயரை அறிமுகப்படுத்துகிறது அதனை ஒரு குறிப்பிட்ட தரவினத்தோடு தொடர்புபடுத்துகிறது
- ✓ ஒரு மாறிக்கு நினைவகத்தில் உரிய இடம் ஒதுக்கப்படும் போது அது வரையறுக்கப்படுகிறது.
- ✓ சில இன மாறிகளை அறிவிக்கும் போதே நினைவகம் ஒதுக்கப்பட்டு விடுகிறது.
- ✓ சுட்டு இன மாறிகள் அறிவிக்கப்படும்போதே வரையறுக்கப்படுவதில்லை. நினைவக ஒதுக்கீடு செய்யப்படும் போதே அவை வரையறுக்கப்படுகின்றன. new செயற்குறி மூலம் நினைவக ஒதுக்கீடு செய்ய முடியும்.

12.4 மதிப்பிருத்து கூற்றுகள் (Assignment Statements)

மதிப்பிருத்து கூற்று, மதிப்பிருத்து செயற்குறிக்கு வலப்புறமுள்ள ஒரு கோவையின் மதிப்பை இடப்புறமுள்ள மாறியில் இருத்துகிறது. = என்பது மதிப்பிருத்து செயற்குறி ஆகும். எடுத்துக்காட்டாக, மாறிகளில் மதிப்பிருத்தும் பல்வேறு வகையான கூற்றுகளைக் காண்க.

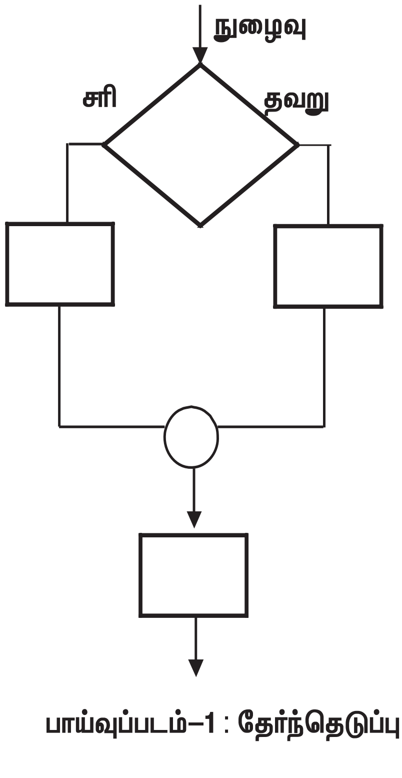
```
num = 5;
total = english+maths;
sum += class_marks;
```

மதிப்பிருத்து செயல்பாட்டின் போது, சி++ நிரல்பெயர்ப்பி, வலப்புறமுள்ள கோவையின் தரவினத்தை, இடப்புறமுள்ள மாறியின் தரவினத்துக்கு மாற்றிக் கொள்கிறது. **11.4.2.** பாடபிரிவில் விளக்கப்பட்ட உள்ளுறை / வெளிப்படை இனமாற்றங்களை நினைவு கூர்க.

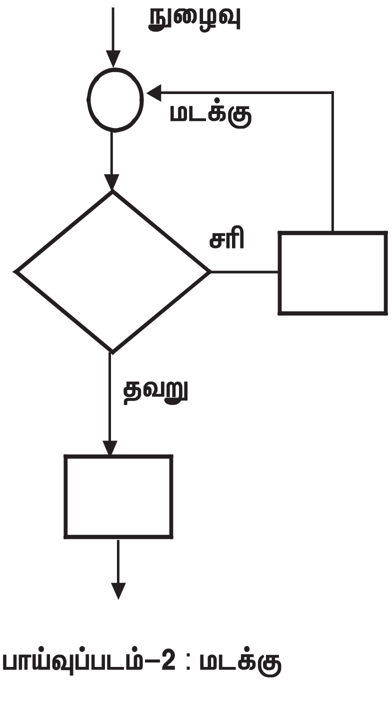
12.5 கட்டுப்பாட்டுக் கட்டளை அமைப்பு (Control Structures)

ஒரு நிரலில் உள்ள கூற்றுகள் கட்டாயமாக ஒன்றன்பின் ஒன்றாய் வரிசையாகத் தான் நிறை வேற்றப்பட வேண்டும் என்கிற தேவை இல்லை. நிரலின் சில பகுதிகள் ஒரு பகுதிகள் ஒரு நிபந்தனையின் பேரில் நிறைவேற்றப்படுவதும் உண்டு. அப்படிப்பட்ட சூழ்நிலையில், பாய்வுக் கட்டுப்பாடு நிரலின் ஒரு பகுதியிலிருந்து இன்னொரு பகுதிக்குத் தாவுகிறது. அத்தகைய தாவலுக்குக் காரணமான நிரல் கூற்றுகள் கட்டுப்பாட்டுக் கூற்றுகள் அல்லது கட்டுப்பாட்டுக் கட்டமைப்புகள் எனப்படுகின்றன. கீழேயுள்ள பாய்வுப் படங்களை நோக்குக:

1. ஒரு முழுஎண்ணை உள்ளீடாகப் பெற்று, அது ஒற்றைப்படை எனில் அதனோடு 1 கூட்டியும், இரட்டைப்படை எனில் அதே எண்ணையும் வெளியிடும் படிநிலைகளைப் பாய்வுப் படம் 1 ல் காண்க. செயல்பாடுகள் ஒன்றன்பின் ஒன்றாய் வரிசையாக நிறைவேற்றப்படுகின்றன.



பாய்வுப்படம்–1 : தேர்ந்தெடுப்பு

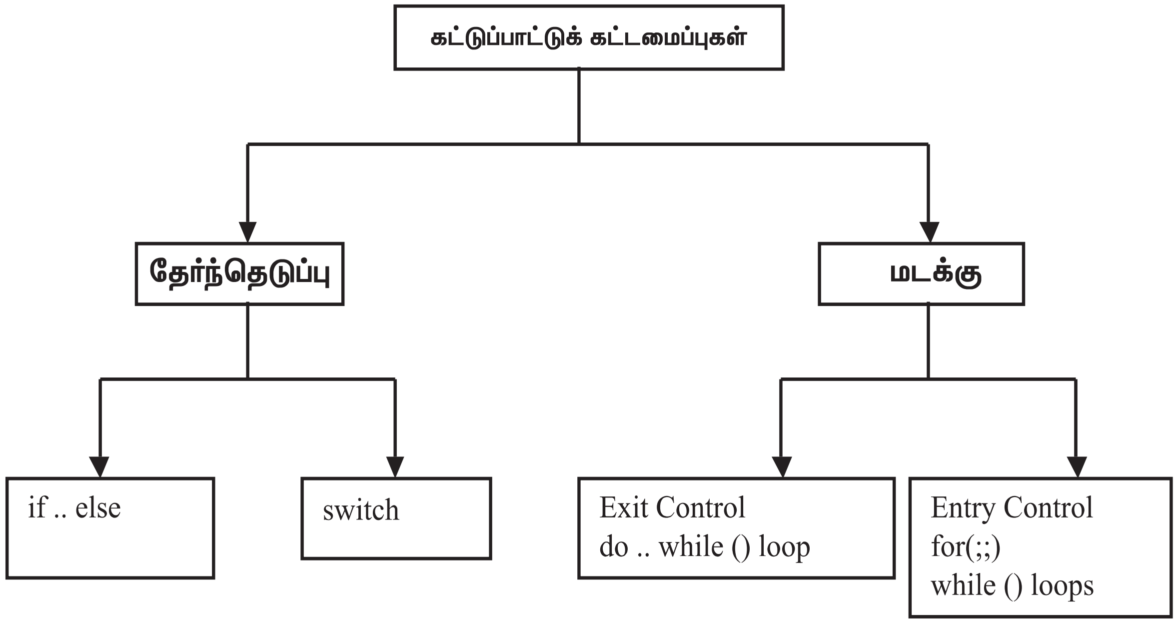


பாய்வுப்படம்–2 : மடக்கு

2. ஒரு முழு எண்ணை உள்ளீடாகப் பெற்று, அது இரண்டால் வகுபடுகிறதா என்பதைச் சோதித்து, அதனடிப்படையில் இரட்டைப்படை அல்லது ஒற்றைப்படை என்னும் செய்தியைக் காட்டும் படிநிலைகளை நோக்குக. இதில் வகுக்கும் போது மீதி வருகிறது எனில் M=ODD என்ற கூற்றுக்கும், மீதி வரவில்லை எனில் M=EVEN என்ற கூற்றுக்கும் பாய்வின் கட்டுப்பாடு கிளை பிரிகிறது.

✓ கட்டுப்பாடு, நிரலின் ஒரு பகுதியிலிருந்து இன்னொரு பகுதிக்குத் தாவுவதற்குக் காரணமான நிரல் சுற்றுகள் கட்டுப்பாட்டுக் கட்டமைப்புகள் எனப்படுகின்றன.

கட்டுப்பாட்டுக் கட்டளை அமைப்புகளுள் இரண்டு முக்கிய வகையினங்கள், தீர்மானிப்புக் கூற்றுகளும் (Decision Making Statements), மடக்குக் கூற்றுகளும் (Looping Statements) ஆகும். தீர்மானிப்புக் கூற்றுகளைத் தேர்ந்தெடுப்புக் கூற்றுகள் (Selection Statements) என்றும் கூறலாம். சி++ மொழியில் கட்டுப்பாட்டுக் கட்டமைப்புகள் கீழேயுள்ள படத்தில் கண்டுள்ளவாறு கட்டுப்பாட்டுக் கூற்றுகளால் நடைமுறைப்படுத்தப்படுகின்றன.



12.5.1 தேர்ந்தெடுப்புக் கூற்றுகள் (Selection Statements)

ஒரு நிரலில் நிபந்தனை அடிப்படையிலான தீர்மானிப்பு (Decision), நிரலின் வேறு பகுதிக்கு **ஒருமுறைத் தாவல்** நடைபெறக் காரணமாகிறது. சி++ மொழியில் தீர்மானிப்புகள் பல்வேறு முறைகளில் மேற்கொள்ளப்படுகின்றன. அவற்றுள் மிகவும் முக்கியமானது. இரண்டு மாற்று வழிகளுள் ஒன்றைத் தேர்ந்தெடுக்கின்ற if ...... else... கூற்றாகும். இன்னொரு தீர்மானிப்புக் கூற்றான Switch, ஒரு மாறியின் மதிப்பு அடிப்படையில் பல்வேறு கிளை பிரிவுகளுள் ஒரு கிளைப் பிரிவிலுள்ள கட்டளைத் தொகுதியை நிறைவேற்ற வழிவகுக்குகிறது.

if **கூற்று :** அனைத்துத் தீர்மானிப்புக் கூற்றுகளிலும் மிகவும் எளிமையானது. இது இரு வடிவங்களில் நடைமுறைப்படுத்தப்படுகிறது.

- if கூற்று (ஒருவழி)
- if..else கூற்று (இருவழி)

```
if (நிபந்தனை/கோவை)
{
 // செயல்பாட்டுத் தொகுதி
}
```

```
if (நிபந்தனை/கோவை)
{
    // செயல்பாட்டுத் தொகுதி–1
}
else
{
    // செயல்பாட்டுத் தொகுதி–2
}
```

கீழேயுள்ள நிரல் (Program 12.2) if கூற்றை விளக்குகிறது

```
// Program - 12.2
# include <iostream.h>
# include <conio.h>
// Demonstrates the use and syntax of if statement
void main ()
{
int a;
clrscr();
cout << "\nEnter a number ";
cin >> a;
if ( a%2 == 0)
cout << "\nThe given number " << a << "is even";
getch();
}
```

மேற்கண்ட நிரலில், தரப்பட்ட நிபந்தனை சரி என்றால் மட்டுமே. "The Given…………..“ என்னும் செய்தி, காட்டப்படும். இல்லையேல் நிரலின் கட்டுப்பாடு cout<<"\n The given“ என்னும் கட்டளையை நிறைவேற்றாமல் அதனைக் கடந்து getch() கூற்றுக்குத் தாவிவிடும்.

கீழேயுள்ள நிரல் (Programme 12.3) if else....கூற்றை விளக்குகிறது.

```
// Program – 12.3
// Demonstrates  the use and syntax of if else statement
# include <iostream.h>
# include <conio.h>
void main()
{
int a;
clrscr();
cout << “\nEnter a number “;
cin >> a;
if ( a%2 == 0)
cout << “\nThe given number “ << a << “is even”;
else
cout << “\nThe given number “ << a << “is odd”;
getch();
}
```

மேற்கண்ட நிரலில், கொடுக்கப்பட்ட கோவை சரி என்றால் மட்டுமே. "The given number 10 is even“ என்னும் செய்தி காட்டப்படும். இல்லையேல் else-க்கு அடுத்துள்ள கூற்று செயல்படுத்தப்படும்.

If கட்டளை அமைப்பில் நிபந்தனைகள் / கோவைகள் வெவ்வேறு பாணியில் தரப்பட்டுள்ள எடுத்துக்காட்டுகளை நோக்குக.

Branch என்னும் மாறி நிபந்தனையாகப் பயன்படுத்தப்பட்டுள்ளது.

```
int branch = 10 > 20;
if (branch )
{
    //செயல்பாட்டுத் தொகுதி–1;
}
else
{
    //செயல்பாட்டுத் தொகுதி–2;
}
```

சுழியம் (zero) அல்லாத எந்தவோர் எண்ணும் சரி என எடுத்துக் கொள்ளப்படும் என்பதால், if **(1)** என்று நிபந்தனையை அமைக்க முடியும்.

```
if (1)
{
    //செயல்பாட்டுத் தொகுதி–1;
}
else
{
    //செயல்பாட்டுத் தொகுதி–2;
}
```

ஒரு கோவையை நிபந்தனையாகத் தரமுடியும். கோவையின் மதிப்பு >**0**–ஆக மதிப்பிடப்படுமாயின், செயல்பாட்டுத் தொகுதி–**1** நிறைவேற்றப்படும். இல்லையேல் செயல்பாட்டுத் தொகுதி–**2** நிறைவேற்றப்படும்.

```
if ( a % 2 )
{
    // செயல்பாட்டுத் தொகுதி–1;
}
else
{
    // செயல்பாட்டுத் தொகுதி–2;
}
```

கீழேயுள்ள நிரலை இயக்கினால் என்ன செய்தி காட்டப்படும் என்பதை முன்னறிந்து சொல்ல இயலுமா **?**

```
// Program - 12.4
# include <iostream.h>
# include <conio.h>
void main()
{
int count = 1;
if (count > 0)
{
cout << “\nNegating count ....”;
count *= -1;
}
else
{
cout << “\nResetting count ...”;
count *= 1;
}
getch();
}
```

இந்த நிரலை இயக்கினால் Negating Count என்னும் செய்திகாட்டப்படும். Count என்னும் மதிப்பு **–1**ஆல் பெருக்கப்பட்ட பின்னும் else க்கு பிறகுள்ள கட்டளைத் தொகுதி ஏன் செயல்படுத்தப்படவில்லை என்பதைச் சொல்ல முடியுமா **?**

இதற்கான பதில் ஒரு if.....else கட்டளை அமைப்பில் ifன் கீழுள்ள கட்டளைத் தொகுதி நிறைவேற்றப்படுமாயின், else தொகுதி நிறைவேற்றப்படாது. If கட்டளைத் தொகுதி நிறைவேற்றப்படாத போது தான் else கட்டளைத் தொகுதி நிறைவேற்றப்படும்.

✓ இரு மாற்று வழிகளுள் ஒன்றைத் தேர்ந்தெடுக்கும் if else கட்டளை அமைப்பில் நிபந்தனை அடிப்படையில் ஒரு குறிப்பிட்ட கட்டளைத் தொகுதி மட்டுமே நிறைவேற்றப்படும்.

வ. எண்	பிழையான கட்டளை அமைப்பு	என்ன காரணம் ?
1.	if a > b cout << "True";	நிபந்தனையைப் பிறை அடைப்புக்குறிகளுக்குள் தர வேண்டும். சரியான கட்டளை **if (a > b)** cout << "True";

2.	if(a > b)
	a - - ;
	cout << "\n Variable is decremented";
	else
	a + +;
	cout << "Variable is incremented ...."
		நிரல் பெயர்ப்பி, "Misplaced else" என்னும் பிழைசுட்டும் செய்தியைத் தரும். செயல்பாட்டுத் தொகுதி ஒன்றுக்கு மேற்பட்ட கட்டளைகளைக் கொண்டிருப்பின் நெளிவு அடைப்புக்குறிகளுக்குள் தர வேண்டும். சரியான கட்டளை அமைப்பு:
		if (a > b)
		{
		a - - ;
		cout<<"\n Variable is decremented";
		}
		else
		{
		a + + ;
		cout << "Variable is incremented.."
		}

3.	if (a > b);
	cout << " Greater..";
	else
	cout << "Lesser..";
		நிபந்தனைக்குப் பிறகு தரப்பட்டுள்ள அரைப்புள்ளி, கை கூற்றின் நோக்கத்தையே தோற்கடித்துவிடும். நிரல்பெயர்ப்பி, எனப் பிழைசுட்டும். சரியான வடிவம் :
		if(a > b)
		cout << "Greater...";
		else
		cout <<"Lesser..";

அட்டவணை 12.1 if **கட்டளை அமைப்பு**

அட்டவணை 12.2.2.ல் தரப்பட்டுள்ள பணிகளை நிறைவேற்றுவதற்குப் பொருத்தமான if கட்டளை அமைப்பை எழுதிக் காட்டுக.

வ.எண்.	பணி	If கட்டளை
1.	மதிப்பெண் 90க்கு மேல் இருப்பின் ‘A’ என்னும் தரநிலை தருக	
2.	மதிப்பெண் 90க்கு மேல் இருப்பின் ‘A’ என்னும் தரநிலை தருக. இல்லையேல் ‘B’ தருக.	
3.	வேகம் மணிக்கு 30 கி.மீ. க்கு குறைவு எனில் ‘Accelerate-traffic to flow’ வேகம் மணிக்கு 31 லிருந்து 40 கி.மீ. வரை எனில் ‘Moderate-Accelerate by 10 kmph’ இல்லையேல், ’Good-be careful…..’ என்ற செய்திகளைத் திரையிட வேண்டும்	

அட்டவணை 12.2 if கட்டளையைப் பயன்படுத்துதல்

பின்னலான if கூற்று– ஒரு if...else கட்டளை அமைப்பிற்குள் இன்னோர் if கட்டளை அமைய முடியும். கிழேயுள்ளவாறு if ....else கட்டளை ஒன்றுக்குள் ஒன்றாய்ப் பின்னலாக அமையலாம்.

```
if (கோவை–1)
   if (கோவை–2)
   {
      செயல்பாடு–1 ;
   }
   else
   {
      செயல்பாடு–2 ;
   }
else
{
   செயல்பாடு–3 ;
};
```

இதுபோன்று பின்னலாய் அமையும். if ....else கட்டளை அமைப்பில் **‘ஒவ்வோர் else –க்கும் மேலாக, வேறோர் else உடன் இணைசேராத ஓர் if இருக்க வேண்டும்.“**

எடுத்துக்காட்டாக,

```
if (grade = = ‘A’)
if (basic > 5500)
incentive = basic * 10/100;
else
incentive = basic * 5/100;
else
cout << “Try to attain Grade A”;
```

மேலேயுள்ள கட்டளைத் தொகுதி செயல்படும் முறை :

- grade == ‘A’, basic ==5501 எனில் incentive–ன் மதிப்பு 550 ஆகும்.
- grade == ‘A’, basic ==5000 எனில் incentive–ன் மதிப்பு 250 ஆகும்.
- grade ! = ‘A’, எனில் உள் if நிறைவேற்றப்படாது. வெளி else நிறைவேற்றப்படும். எனவே “Try to attain Grade ‘A’ என்னும் செய்தி கிடைக்கும்.

மேலேயுள்ள if கட்டளை அமைப்பு, கீழேயுள்ள if கட்டளை அமைப்புக்கு நிகரானதா ? உங்கள் பதிலை ‘காரணம் கூறுக’ பெட்டியில் எழுதுக.

```
if (grade = = ‘A’  && basic > 5500)
        incentive = basic * 10/100;
else if (grade = = ‘A’ && basic <5501)
         incentive = basic * 5/100;
else
    cout << “Try to attain Grade A”;
```

காரணம் கூறுக.........

Switch கூற்று : இது ஒரு கிளைபிரிப்புக் கூற்றாகும். ஒரு நிபந்தனையின் அடிப்படையில், தரப்பட்டுள்ள பல்வேறு தேர்வுகளில் ஒன்றுக்குக் கட்டுப்பாட்டை எடுத்துச் செல்லும். இக்கட்டளை இவ்வாறு நடைமுறைப்படுத்தப்படுகிறது.

```
switch (கோவை)
{
 case 1 :  செயல்பாட்டுத் தொகுதி–1;
                 break;
 case 2 :  செயல்பாட்டுத் தொகுதி–2;
                 break;
 case 3 :  செயல்பாட்டுத் தொகுதி–3;
                 break;
 default:  செயல்பாட்டுத் தொகுதி–4;
}
```

```
switch (remainder)
{
   case  1 : cout << “remanider 1”;
             break;
   case  2 : cout << “remanider 2”;
             break;

    default :
             cout << “Divisible by 3”;
}
```

கீழேயுள்ள நிரல், Switch கூற்றின் பயன்பாட்டை விளக்குகிறது.

```
// Program - 12.5
// to demonstrate the use of switch statement

# include <iostream.h>
# include <conio.h>

void main()
{
int a, remainder;
cout << “\nEnter a number ...”;
cin >> a;
```

```
 remainder = a % 3;
switch (remainder)
{
case 1 : cout << "\nRemainder is one"; break;
case 2 : cout << "\nRemainder is two"; break;
default: cout << "\nThe given number is divisible by 3";
break;
}
getch();
}
```

மேற்கண்ட நிரல்,

- a=4 அல்லது அதுபோன்று மீதி **1** வரும் எண் எனில், 'Remainder is one' எனக் காட்டும்.
- a=5 அல்லது அதுபோன்று மீதி **2** வரும் எண் எனில், 'Remainder is two' எனக் காட்டும்.
- a=3 அல்லது அதுபோன்று **3–**ஆல் வகுபடும் எண் எனில், 'The given number is divisable by 3' எனக் காட்டும்.

சுருங்கச் சொல்வதெனில், கொடுக்கப்பட்ட எண் மூன்றால் வகுபடும் எண்ணா எனப் பார்த்து அதற்கேற்ப செய்தியைத் திரையில் காட்டும்.

கீழேயுள்ள நிரலின் வெளியீடு என்னவாக இருக்கும் என்று ஊகிக்க முடிகிறதா ?

```
// Program - 12.6
// to demonstrate the use of switch statement
# include <iostream.h>
# include <conio.h>
void main()
{
int rank = 1;
char name[ ] = "Shiv";
switch (rank)
{
case 1 : cout << '\n' << name << " secured 1st rank";
case 2 : cout << '\n' << name << " secured 2nd rank";
}
getch();
}
```

இந்த நிரலின் வெளியீடு இவ்வாறு இருக்கும் :

Shiv secured 1st rank

Shiv secured 2nd rank

case1, case2 ஆகிய இரண்டு செயல்பாட்டுத் தொகுதிகளுமே ஏன் செயல்படுத்தப்பட்டுள்ளது என்பது புரிகிறதா? Program–12.5 மற்றும் Programme–12.6 ஆகிய இரு நிரல்களிலும் உள்ள செயல்பாட்டுத் தொகுதிகளை ஒப்பிட்டுப் பாருங்கள். Programme–12.6–ல் விடுபட்டுப்போன கட்டளை எதுவெனத் தெரிகிறதா? ஆம் Break கட்டளை விடுபட்டுள்ளது. இதிலிருந்து அறிய வருவது என்ன?

ஒவ்வொரு செயல்பாட்டுத் தொகுதியிலும் இறுதியில் break கட்டளை இடம் பெற வேண்டும். இல்லையேல், நிபந்தனையின் அடிப்படையில் கட்டுப்பாடு தாவிச் சென்று நிற்கும் Case–ல் தொடங்கி அடுத்தடுத்துள்ள Case–களின் செயல்பாட்டுத் தொகுதிகள் வரிசையாக நிறைவேற்றப்படும்.

மேற்கண்ட எடுத்துக்காட்டில் (Program–12.6) கட்டுப்பாட்டு Case1க்கு எடுத்துச் செல்லப்படுகிறது. அதன் செயல்பாட்டுத் தொகுதி நிறைவேற்றப்படுகிறது. அதில் break கட்டளை இல்லாத காரணத்தால் case 2ன் செயல்பாட்டுத் தொகுதியும் வரிசையாக நிறைவேற்றப்படுகிறது.

✓ குறிப்பிட்ட case–ன் செயல்பாட்டுத் தொகுதி நிறைவேற்றப்பட்டவுடன் switch கூற்றைவிட்டு வெளியேற, அத்தொகுதியின் இறுதியில் break கட்டளையைச் சேர்க்க வேண்டும்.

கீழே காணும் Switch கட்டளை அமைப்புகள் பிழையானவை. காரணம் காண்க.

1.

```
char name[ ] = “Shiv”;
switch (name)
{
case “Shiv” : cout << ‘\n’
<< name << “ secured 1st rank”;
case “Rama” : cout << ‘\n’
<< name << “ secured 2nd rank”;
}
```

" switch selection expression must be of integral type" என நிரல் பெயர்ப்பி பிழைசுட்டும். இதன் பொருள், switch–ல் இடம் பெறும் கோவையின் மதிப்பு முழு எண் (char, enum, int) இனம் சார்ந்த மாறிலியாக இருக்க வேண்டும்

2.

```
float value;
switch (value)
{
case 1.5 : cout << ‘\n’
<< value – 0.5;
case 2.9 : cout << ‘\n’
<< value + 0.1;
}
```

Switch() கூற்றில் இடம்பெறும் கோவையின் float இனமாக இருப்பதால் பிழையாகிறது

3. switch (rank)
{
case 1 to 2 : cout << '\n'
<< "Best rank";
break;
case 3 to 4 : cout << '\n'
<< "Good rank";
}

Case 1 to 2 என்பது ஏற்கத் தகாத கூற்றாகும். Case–ல் ஒரேயொரு முழுஎண் மதிப்பு மட்டுமே இடம் பெற வேண்டும். ஒரு குறிப்பிட்ட செயல்பாட்டுத் தொகுதியை ஒன்றுக்கு மேற்பட்ட மதிப்புகளுக்குப் பயன்படுத்த வேண்டுமெனில், இந்த நிரல் பகுதியை இவ்வாறு எழுதவேண்டும்.

switch (rank)

{

case 1 :

case 2 : cout << "Best

rank"; break;

case 3 :

case 4 : cout << "Good

rank"; break;

}

12.5.2. மடக்குகள் (Loops)

மடக்குகள், ஒரு கட்டளைத் தொகுதியை குறிப்பிட்ட தடவைகள் திரும்பத் திரும்ப நிறைவேற்றுகின்றன. எடுத்துக்காட்டாக, கீழேயுள்ள Programme–12.7–ஐ நோக்குக.

```
// Program - 12.7
# include <iostream.h>
# include <conio.h>

void main()
{
  int i = 1;
   loop_start:
   if (i < 6)
   {
              cout << i << '\t';

              i = i + 1;
                goto loop_start;
   }
}
```

கட்டளைத் தொகுதியை நிறைவேற்றுவதற்குப் பரிசோதிக்க வேண்டிய நிபந்தனை. (if (i < 6))

திரும்பத் திரும்ப நிறைவேற்றப்பட வேண்டிய கட்டளைத் தொகுதி

நிரலின் கட்டுப்பாடு, திரும்பத் திரும்ப நிறைவேற்றப்பட வேண்டிய கட்டளைத் தொகுதியின் தொடக்கத்துக்கு எடுத்துச் செல்லப்படும். (goto loop_start;)

மேலே காணும் நிரலை இயக்கினால், if கூற்றின் கீழ் தரப்பட்டுள்ள செயல்பாட்டுத் தொகுதி 5 முறை நிறைவேற்றப்பட்டு, 1 முதல் 5 வரையிலான எண்களைத் திரையில் காட்டும்.

Program– 12.7 இவ்வாறு செயல்படுகிறது

1. i என்னும் மாறி அறிவிக்கப்பட்டு தொடக்க மதிப்பிருத்தப்படுகிறது.
2. i<6 என்னும் ஒப்பீட்டுக் கோவை பரிசோதிக்கப்படுகின்றது.
3. நிபந்தனை சரி எனில் செயல்பாட்டுத் தொகுதி (cout<<i;i=1;) நிறைவேற்றப்படுகிறது. நிரலின் கட்டுப்பாடு மீண்டும் loop_start என்னும் இடத்துக்கு எடுத்துச் செல்லப்படுகிறது (goto loop_start), ஒப்பீட்டுக் கோவையின் நிபந்தனை அடிப்படையில் ஒரு கட்டளைத் தொகுதி திரும்பத் திரும்பச் செயல்படுத்தப்படுகிறது. i–என்ற மாறி, கட்டுப்பாட்டு மாறி எனக் குறிப்பிடப்படுகிறது. காரணம், கட்டளைத் தொகுதி திரும்பத் திரும்ப நிறைவேற்றப்பட i–ன் மதிப்பே காரணமாக அமைகின்றது.

ஆக, மடக்குத் தொகுதி, உடற்பகுதி, கட்டுப்பாட்டுக் கூற்று என்னும் இரு பகுதிகளைக் கொண்டுள்ளது. கட்டுப்பாட்டுக் கூற்று நிபந்தனையைச் சரிபார்க்கிறது. அதனடிப்படையில், நிரலின் கட்டுப்பாடு, மடக்கின் உடற்பகுதிக்கு மீண்டும் மீண்டும் திருப்பிவிடப்பட்டு, திரும்பத் திரும்ப நிறைவேற்றப்படுகிறது. இப்போது கீழேயுள்ள நிரல்களை நோக்குங்கள்.

//Program - 12.7 A	//Program - 12.7 B
void main()	void main()
{	{
int i = 6;	int i = 6;
loop_start:	**loop_start:**
if (i < 6)	{
{	cout << i << '\t'; i = i + 1;
cout << i << '\t'; i = i + 1;	if (i < 6)
goto loop_start;	**goto** loop_start;
}	}
cout << i;	cout << i;
}	}

மேற்கண்ட நிரல்களின் வெளியீடு என்னவாக இருக்குமென்று நினைக்கிறீர்கள் ?

Program -12.7 A, 6 எனத் திரையில் காட்டும். Program -12.7B, 7 எனக்காட்டும். இந்த நிரலில், (cout << i; i = i + 1;) ஆகிய கட்டளைகளுக்குப் பிறகு நிபந்தனை இடம்பெற்றுள்ளது. எனவே இக்கட்டளைகள் ஒருமுறை நிறைவேற்றப்பட்டு விடும். அதன்பின் சரிபார்க்கப்படுகிறது. இப்போது i–ன் மதிப்பு 7 என்பதால் நிரலில் கட்டுப்பாடு loop_start. இருக்குமிடத்துக்கு எடுத்துச் செல்லப்படவில்லை. இதிலிருந்து நாம் அறிவது என்ன?

✓ மடக்கில் உடற்பகுதியின் இறுதியில் நிபந்தனை தரப்படுமாயின் நிறைவேற்றப்பட்டுவிடும்.

✓ நிபந்தனை இடம்பெறும் இடத்தைப் பொறுத்து, மடக்குகளை, நுழைவு–சோதிப்பு மடக்கு (Program-12.7 A), வெளிறேல்–சோதிப்பு மடக்கு (Program-12.7 B) என இருவகையாகப் பிரிக்கலாம்.

பொதுவாக ஒரு மடக்கின் செயலாக்கம், இவ்வாறு நடைபெறுகிறது.

1. நிபந்தனை மாறியில் தொடக்க மதிப்பு இருத்தப்படுகிறது.
2. மடக்கின் உடற்பகுதி ஒருமுறை நிறைவேற்றப்படுகிறது.
3. நிபந்தனை மாறியின் மதிப்பு மிகுக்கப்படுகிறது
4. நிபந்தனை மாறி, ஓர் ஒப்பீட்டுக் கோவையில் பரிசோதிக்கப்படுகிறது. ஒப்பிட்டுக் கோவையின் மதிப்பு அடிப்படையில் கட்டுப்பாடு, உடற்பகுதியின் தொடக்கத்துக்கு எடுத்துச் செல்லப்படும் அல்லது மடக்கை விட்டு வெளியேறும்.

சி++ மொழியில், மூன்று வகையான மடக்குகள் உள்ளன.

1. for() மடக்கு
2. while() மடக்கு
3. do....while() மடக்கு

do……while () மடக்கு :

இதன் கட்டளை அமைப்பு இவ்வாறு இருக்கும்.

```
do
{
செயல்பாட்டுத் தொகுதி;
} while <(நிபந்தனை)>;
```

கீழேயுள்ள நிரலை நோக்குங்கள்

```
// Program - 12.8
# include <iostream.h>
# include <conio.h>
// to print the square of numbers
// between 2 to 5
void main()
{
clrscr();
int num = 2;
do
{
cout << num * num << '\t';
num += 1;
}
while (num < 6);
getch();
}
```

Program - 12.8 நிரலின் அடிப்படையில் கீழேயுள்ள வினாக்களுக்கு விடை தருக.

அ) அடையாளம் காண்க.

1. பயன்படுத்தப்பட்டுள்ள கட்டுப்பாட்டு மாறி

2. மடக்கின் உடல்பகுதியிலுள்ள கட்டளைகள்

3. பரிசோதிப்புக் கோவை

ஆ) மடக்குக் கட்டளைகள் எத்தனை முறை நிறைவேற்றப்படும் ?

இ) நிரலின் வெளியீடு என்ன ?

ஈ) இது என்ன வகையான மடக்கு ?

விடைகள்

அ) 1. கட்டுப்பாட்டு மாறி num

2. மடக்கின் உடல்பகுதிக் கட்டளைகள்

```
cout<<num*num<<'\t';
num+=1;
```

3. (num<6) என்பது பரிசோதிப்புக் கோவை

ஆ) நான்கு முறை

இ) 4 9 16 25

ஈ) வெளியேறல் – சோதிப்பு மடக்கு

இந்த மடக்கின் செயல்பாட்டுப் படிநிலைகளைக் காண்க.

1. மடக்கில் நுழைகிறது.
2. num- இரண்டாம் அடுக்கு மதிப்பைத் திரையில் காட்டுகிறது.
3. கட்டுப்பாட்டு மாறியின் மதிப்பு 1 மிகுக்கப்படுகிறது.
4. நிபந்தனை சோதிக்கப்படுகிறது. சோதனை முடிவின் அடிப்படையில் கட்டுப்பாடு படிநிலை 2க்கு மாற்றப்படுகிறது.
5. மடக்கு முடிவு பெறுகிறது.

do...while (நிபந்தனை) என்னும் மடக்கு வெளியேறல் – சோதிப்பு மடக்கு எனப்படும். காரணம், நிபந்தனைக் கோவை, மடக்கின் உடற்பகுதிக் கட்டளைகளுள் இறுதியாக இடம் பெற்றுள்ளது. நிபந்தனை கோவையைப் பல்வேறு வகையில் அமைக்க முடியும். எடுத்துக்காட்டுகள் காண்க.

```
int ctr = 1, sum = 0, check = 1;
do
{
  cout << ctr;
   sum = sum + ctr;
   ctr = ctr + 2;
   check = (ctr < 11);
}while(check);
```

```
int ctr = 5, sum = 0;
do
{

  cout << ctr;
   sum = sum + ctr;
   ctr = ctr -  2;

}while(ctr);
```

```
int ctr = 5,sum = 0,c=1;
do
{
cout << ctr;
sum = sum + ctr;
ctr = ctr - 2;


} while(ctr >= 1);
```

கீழேயுள்ள நிரல்களின் வெளியீடு என்னவாக இருக்கும்?

```
// snippet A
# include <iostream.h>
# include <conio.h>

void main()
{
   int i = 10;
   do
   {
       cout << i;
       i--;
   } while (i <= 10);
   getch();
}
```

```
//snippet B
# include <iostream.h>
# include <conio.h>

void main()
{
  int i = 10; choice = 1;
  do
  {
    cout << i;
    i++;
  }while (choice);
  getch();
}
```

Snippet A – யில் i–ன் மதிப்பு **–32768** ஆகும்வரை மடக்கு நிறைவேற்றப்படும். Snippet A. Snippet B ஆகிய இரண்டு நிரல்களிலும் உள்ள மடக்கு வரம்பிலா மடக்குகளாகும். இரண்டு மடக்குகளிலும் தரப்பட்டுள்ள நிபந்தனைகள் எப்போதுமே சரியாகவே இருக்கும். Snippet A-யில் i–ன் மதிப்பு எப்போதுமே **10**–ஐ விடக் குறைவாகவே இருக்கும். Snippet B-யில் choice-ன் மதிப்பு எப்போதுமே **1** என்பதால் நிபந்தனை எப்போதுமே சரி என்றாகி விடுகிறது. ஒரு கட்டத்தில் நிபந்தனை ‘தவறு’ என்று ஆகுமாறு சரியான நிபந்தனையை அமைப்பது மிகவும்

முக்கியமாகும். மடக்கில் தரப்படும் சோதிப்புக் கோவையோடு தொடர்புடைய கட்டுப்பாட்டு மாறியின் மதிப்பை மிகுத்து/புதுப்பித்து இந்த நிலையை எட்ட வேண்டும்.

while () மடக்கு :

இது நுழைவு சோதிப்பு மடக்கு எனப்படும். இதன் கட்டளை அமைப்பு

```
while <(condition)>
{
செயல்பாட்டுத் தொகுதி;
}
```

while() கூற்றில் உள்ள சோதிப்புக் கோவை 'சரி' என்னும் விடையைத் தந்தால் மட்டுமே மடக்கின் உடற்பகுதியிலுள்ள கட்டளைகள் நிறைவேற்றப்படும். சோதிப்புக் கோவையின் மதிப்பு 'தவறு' என ஆகும் போது மடக்கு முடிவுக்கு வரும். do....while மடக்கில் விளக்கப்பட்ட அதே நிரலை while() மடக்கு பயன்படுத்தி மீண்டும் எழுதுவோம். (Program - 12.9)

```
// Program - 12.9
# include <iostream.h>
# include <conio.h>

// to print the square of numbers
// between 2 to 5

void main()
{
  clrscr();
  int num = 2;
  while (num < 6)
  {
        cout << num * num << '\t';
        num += 1;
  }
  getch();
}
```

மடக்கின் உடற்பகுதியின் தொடக்கத்தில் சோதிப்புக் கோவை அதாவது நிபந்தனை தரப்பட்டுள்ளது.

மேற்கண்ட மடக்கு இவ்வாறு செயல்படும்.

1. num என்னும் கட்டுப்பாட்டு மாறியில் 2 என்னும் தொடக்க மதிப்பு இருத்தப்படுகிறது.
2. num<2 என்னும் சோதிப்புக் கோவை மதிப்பிடப்பட்டு, விடை 'சரி' என்றால் மட்டுமே கட்டுப்பாடு படிநிலை 3க்கு மாற்றப்படுகிறது.
3. num என்னும் மாறியில் இருத்தப்பட்ட மதிப்பின் இரண்டாம் அடுக்கு, திரையில் காட்டப்படுகிறது.
4. num மாறியின் மதிப்பு ஒன்று மிகுக்கப்படுகிறது.
5. நிரலின் கட்டுப்பாடு படிநிலை 2க்கு மாற்றப்படுகிறது
6. மடக்கு முடிவு பெறுகிறது

மேற்கண்ட நிரல் Program 12.10–ன் அடிப்படையில் கீழ்க்காணும் வினாக்களுக்கு

விடை அளிக்கவும்.

```
//Program-12.10
# include <iostream.h>
# include <conio.h>
void main()
{
int x = 3, y = 4, ctr = 2,res = x;
while(ctr <= y)
{
res *= x;
ctr += 1;
}
cout << “x to the power of y is : ” << res;
getch();
}
```

அ) 1. இந்த நிரலில் பயன்படுத்தப்பட்டுள்ள கட்டுப்பாட்டு மாறி எது ?

2. மடக்கின் உடற்பகுதிக் கட்டளைகள் எவை ?

3. பரிசோதிப்புக் கோவை எது ?

ஆ) மடக்கு எத்தனை முறை நிறைவேற்றப்படுகிறது ?

இ) இந்த நிரலின் வெளிப்பாடு யாது ?

ஈ) இதில் பயன்படுத்தப்பட்டுள்ளது எந்த வகையான மடக்கு ?

விடைகள்.

அ) 1. கட்டுப்பாட்டு மாறி ctr

2. res *= x; ct +=1;

3. ctr <= y

ஆ) மூன்று தடவைகள்

இ) 81

ஈ) ‘நுழைவு சோதிப்பு ‘ அல்லது ‘நுழைவுக் கட்டுப்பாட்டு மடக்கு’ வகை.

கீழேயுள்ள நிரல் Program 12.11ன் வெளியீடு என்னவாக இருக்கும் ? இந்த நிரலுக்கு உள்ளீடாக y,y,y,y,n எனத் தரப்பட்டுள்ளதாக கொள்க.

```
// Program - 12.11
# include <iostream.h>
# include <conio.h>
void main()
{
clrscr();
int counter = 0;
char choice = 'y';
while (choice == 'y')
{
cout << "Continue <y/n> ...";
cin >> choice;
counter = counter + 1;
}
cout << "\The loop is executed .." << counter << " times";
getch();
}
```

கீழேயுள்ள குறுநிரல்கள் பிழையானவை. இவற்றிலுள்ள பிழைகளைக் கண்டறிக. நிரல்கள் சரியாகச் செயல்படுமாறு பிழைகளைத் திருத்துக.

```
/ /Program - 12 A
# include <iostream.h>
# include <conio.h>
//to print numbers between
5&10
void main()
{
  int start = 5,end = 10;
  while (start >= end)
          cout << start++;
  getch();
}
```

```
//Program - 12 B
# include <iostream.h>
# include <conio.h>
//to print numbers between 5&10
void main()
{
   int start = 5,end = 10;
  while (start <= end)
          cout << start++;
  getch();
}
```

```
//Program – 13 A
# include <iostream.h>
# include <conio.h>
// to print numbers between
10&5
void main()
{
   clrscr();
   int start = 5,end = 10;
   while (start <= end)
            cout << start--;
   getch();
}
```

```
//Program – 13 B
# include <iostream.h>
# include <conio.h>
// to print numbers between 10&5
void main()
{
   clrscr();
   int start = 5,end = 10;
   while (start <= end)
            cout << end--;
   getch();
}
```

```
//Program – 14 A
# include <iostream.h>
# include <conio.h>
// to print numbers
          between 1 & 5
void main()
{
   clrscr();
   int start = 1;
   while (Start <=5)
            cout << start++;
   getch();
}
```

```
//Program – 14 B
# include <iostream.h>
# include <conio.h>
// to print numbers
          between 1 & 5
void main()
{
   clrscr();
   int start = 1;
   while (1)
            cout << start++;
   getch();
}
```

for () மடக்கு :

இது ஒரு நுழைவுக் கட்டுப்பாட்டு மடக்கு. ஒரு செயல்பாட்டைக் குறிப்பிட்ட தடவைகள் நிறைவேற்றுவதற்கு இது பயன்படுகிறது. இதன் கட்டளை அமைப்பு

for(தொடக்க மதிப்பு; சோதிப்பு நிபந்தனை; மிகுப்பு)

{

செயல்பாட்டுத் தொகுதி ;

}

for(;;) மடக்குப் பொதுவாக இவ்வாறு செயல்படும்:

1. நிரலின் கட்டுப்பாடு முதன்முதலில் மடக்கினில் நுழையும்போது கட்டுப்பாட்டு மாறியில் தொடக்க மதிப்பு இருத்தப்படும்.

2. நிபந்தனை பரிசோதிக்கப்படும். நிபந்தனை ‘சரி’ எனில் மடக்கின் உடற்பகுதி செயல்படுத்தப்படும். இதன் காரணமாகவே இந்த மடக்கு ‘நுழைவுக் கட்டுப்பாட்டு மடக்கு’ எனப்படுகிறது.

3. மடக்கின் உடற்பகுதி மீண்டும் நிறைவேற்றப்படுவதற்கு முன்பாகக் கட்டுப்பாட்டு மாறியின் மதிப்பு மிகுக்கப்பட்டு, மீண்டும் நிபந்தனை பரிசோதிக்கப்படும்.

4. நிபந்தனை ‘தவறு’ என ஆகும் போது மடக்கு முடிவுக்கு வரும்.

கீழேயுள்ள நிரல் for(;;) மடக்கின் செயல்பாட்டை விளக்கும்.

```
// Program - 12.15
# include <iostream.h>
# include <conio.h>
void main()
{
int i,fact = 1;
for(i = 1; i < 6; i++)
fact *= i;
cout << “\nThe factorial of the number is ..” << fact;
}
```

for(i = 1; i < 6; i++)

- i++ → **மிகுத்தல் (முதல் பகுதி)**
- i < 6 → **பரிசோதிப்பு நிபந்தனை (2ம் பகுதி)**
- i = 1 → **கட்டுப்பாட்டு மாறியில் தொடக்க மதிப்பிருத்தல் (3ம் பகுதி)**

✓ கட்டுப்பாட்டு மாறியில் தொடக்க மதிப்பிருத்தல், மடக்கின் உடற்பகுதி முதன் முதலில் செயல்படுத்தப்படுவதற்கு முன்பாக, ஒரே ஒருமுறை தான் செயல்படுத்தப்படும்.

✓ மடக்கின் உடற்பகுதி ஒவ்வொருமுறை செயல்படுத்தப்படுவதற்கு முன்பாகவும் நிபந்தனை பரிசோதிக்கப்படும்.

✓ நிபந்தனை பரிசோதிக்கப்படுவதற்கு முன்பாகக் கட்டுப்பாட்டு மாறியின் மதிப்பு மிகுக்கப்படும்.

கீழேயுள்ள நிரல்களை நோக்குக. இவற்றின் வெளியீடு என்னவாக இருக்கும் ? எழுதிக் காட்டுக.

```
//  Program – 12.16
# include <iostream.h>
# include <conio.h>

void main()
{
    int ctr = 10;
    for(; ctr >= 6; ctr—)
       cout << ctr << '\n';
}
```

வெளியீடு

```
10
9
8
7
6
```

```
//Program – 12.17
# include <iostream.h>
# include <conio.h>

void main()
{
  clrscr();
  for(int i=2,fact =1;i<6;fact*=i,i++);
  cout << "\nThe factorial  .." << fact;

  getch();
}
```

வெளியீடு

The factorial.. 120

for() கூற்றில் தொடக்க மதிப்பிருத்தல் மற்றும் மிகுப்புப் பகுதியில் ஒன்றுக்கு மேற்பட்ட கூற்றுகள் இடம் பெற்றுள்ளன. கவனித்தீர்களா ? கட்டளை அமைப்பின்படியும் தருக்க முறையிலும் மேற்கண்ட கூற்று சரியானதே மடக்கிலுள்ள ஒவ்வொரு பகுதியும் பல கூற்றுகளைப் பெற்றிருக்க முடியும். அவை காற்புள்ளியால் பிரிக்கப்பட்டிருக்க வேண்டும். கீழேயுள்ள புரோகிராம் பகுதியின் வெளியீடு என்னவாக இருக்கும் ?

```
void main()
{

   for (int i = 1, j = 0 ; i < 8,j<3;i++,j++)
       cout << i << '\t';
  for (int i = 1,,j = 0 ;j < 3,i < 8;i++,j++)
       cout << i << '\t';
}
```

வெளியீடு இவ்வாறு இருக்கும். 1 2 3 // j< 3 வரை மடக்கு செயல்படும் 1 2 3 4 5 6 7//j<8 வரை மடக்குச் செயல்படும் காற்புள்ளிச் செயற்குறி செயல்படும். திசைமுகம் நினைவு கூர்க. ஒரேயொரு நிபந்தனை (வலப்புறம் உள்ளது) மட்டுமே சரிபார்க்கப்படும் ஒன்றுக்கு மேற்பட்ட நிபந்தனைகள் && அல்லது || செயற்குறியுடன் தரப்பட வேண்டும்.

```
// Program – 12.18
# include <iostream.h>
# include <conio.h>

void main()
{
  clrscr();
  int sum =0, ctr = 1;
  for(;ctr <= 5;)
  {
    sum += ctr;
    ctr = ctr + 1;
  }
  cout << "\nSum :" << sum;
  getch();
}
```

வெளியீடு இவ்வாறு இருக்கும்.

```
Sum : 15
```

தொடக்க மதிப்பிருத்தல் மற்றும் மிகுப்புக் கூற்றுகள் for() அமைப்புக்குள் தரப்படவில்லை என்பதைக் கவனித்தீர்களா ?

```
// Program – 12.19
# include <iostream.h>
# include <conio.h>
void main()
{
  clrscr();
  int sum =0, ctr = 1;
  char ch ='y';
  for(;ch == 'y';)
  {
    sum += ctr;
  ctr++;
  cout << "\nContinue <y/n>?..";

    cin >> ch;
  }
  cout << "\nSum :" << sum;
  cout << "\nChoice : " << ch;
  getch();
}
```

```
Continue <y/n> ? ..y
Continue <y/n> ? ..y
Continue <y/n> ? ..y
Continue <y/n> ? ..n
sum:10
Choice : n
```

இதில் for() மடக்கு எத்தனை தடவைகள் நிறைவேற்றப்பட வேண்டும் என்பது இயக்கநேரத்தில் தீர்மானிக்கப்படுவதைக் கவனித்தீர்களா ?

கீழேயுள்ள நிரல் பகுதியில் காணப்படும் பிழை யாது ?

```
int ctr,sum = 0;

for(ctr = 1; ctr < 5; ctr++);

     sum += ctr;

 cout << sum;
```

இதன் வெளியீடு **5** எனக் காட்டும். காரணம் என்னவென்று புரிகிறதா ? for() கூற்றை அடுத்து அரைப்புள்ளி இடப்பட்டுள்ளது. எனவே sum+=ctr என்னும் கட்டளை, மடக்கின் உடற்பகுதியாகக் கருதப்படவில்லை.

கீழேயுள்ள நிரல்பகுதிகளின் வெளியீடு என்னவாக இருக்கும் ?

```
int ctr = 1;
for( ; ctr < 10; ctr++ )
 {
    cout << ctr;
    ctr = 1;
}
```

```
for(it ctr=1;;ctr++)
     cout << ctr;
```

நிபந்தனைக் கோவை இல்லாத காரணத்தால் int இனத்தின் வரம்பெல்லை –32768 +32767 ஆகியவற்றுக்கு இடையேயான மதிப்புகளைத் தொடர்ந்து காட்டிக் கொண்டிருக்கும். ‘மடக்கு முடிவிலா மடக்கு’ ஆகிவிடும்.

12.5.3 continue

ஒரு மடக்கின் உடற்பகுதியில் இடம்பெறும் continue கட்டளை, மடக்கினை அடுத்த சுழற்சிக்கு இட்டுச்செல்லும். மடக்கின் உடற்பகுதியில் continue கட்டளைக்குப்பின் இடம்பெறும் கட்டளைகள் நிறைவேற்றப்படா.

கட்டுப்பாடு for() மடக்கின் மிகுப்புப் பகுதிக்கு எடுத்துச் செல்லப்படும்

```
//Program– 12. 20
# include <iostream.h>
# include <conio.h>

void main()
{
  int i = 1,sum = 0;
  for(;i<10;i++)
  {
          if( i % 2 == 0)
           {
             sum += i;
             continue;
           }
           cout <<i;
   }
   cout << “\n Sum of even  nos..”<<sum;
   getch();
  }
```

பல்வேறு மடக்குகளில் continue செயல்படும் விதத்தைக் கீழே காண்க

```
for( ; ; )
{
   ….
   continue;
    …..
}
```

```
do
{
   ….
   continue;
    ….
}while( )
```

```
while( )
{
   ….
   continue;
    …..
}
```

12.5.4 break

பரிசோதிப்பு நிபந்தனை ‘தவறு’ ஆகும் போது, மடக்கின் செயல்பாடு நிறுத்தப்படுகிறது. ஆனால் சில சூழ்நிலைகளில், பரிசோதிப்பு நிபந்தனை என்னவாக இருந்தாலும் குறிப்பிட்ட தருணத்தில் மடக்கினை முடித்து வைக்க நேரலாம். எடுத்துக்காட்டாக, program 12.21-ஐ நோக்குங்கள்.

```
//Program - 12.21
# include <iostream.h>
# include <conio.h>

void main()
{
clrscr();
int a[] = {1,2,3,4,5,6,7,8,9};
int search_item = 7;
for(int x=0; x<9;x++)
{
        if (a[x] == search_item)
        {
          cout << “\nItem found at position ..” << x; break;
          }
}
cout << ‘\n’ << “value of index position is ..” << x;
getch();
}
```

இதன் வெளியீடு இவ்வாறு அமையும்

```
Item found at position .. 6
value of index position is .. 6
```

✓ break கட்டளை, நிரலின் கட்டுப்பாட்டை மடக்குக்கு வெளியே எடுத்துச் செல்கிறது. x-ன் மதிப்பு 6–ஆக இருக்கும்போது, break கட்டளை மடக்கினை முறிக்கிறது.

✓ break கட்டளை இது இடம்பெறும் மடக்கினைவிட்டு வெளியேறச் செய்யும்.

✓break கட்டளை இது இடம்பெறும் மடக்கினைவிட்டு தாவச் செய்யும்.

பின்னலான மடக்குகள் (Nested Loops)

ஒரு மடக்கின் உடற்பகுதியில் இன்னொரு மடக்கு பின்னலாய் அமையலாம். கீழேயுள்ள நிரல் Program 12.22ஐ நோக்குக.

```
// nesting of loops – Program- 12.22
# include <iostream.h>
# include <conio.h>

void main()
{
  clrscr();
  for(int i = 1; i <= 3; i++)
  {
          int j = 1;
          while(j <= i)
          {
            cout << "* ";
            j++;
          }
          cout << '\n';
  }
  getch();

}
```

இதன் வெளியீடு

```
*
* *
* * *
```

மடக்குகளின் செயல்பாடு இவ்வாறாக அமைகிறது

```
for(int i = 1; i <= 3; i++)
 {
   int j = 1;
   while(j <= i)
   {
     cout << "* ";
     j++;
   }
   cout << '\n';
}
```

பின்னல் மடக்குகளின் மடக்குச் செயல் இவ்வாறு அமைகிறது

for ..loop	while loop
i = 1	ஒருமுறை நிறைவேற்றப்படும் 1 (j<= I)
i = 2	இருமுறை நிறைவேற்றப்படும் (j = 1 .. 2)
i = 3	மும்முறை நிறைவேற்றப்படும் (j = 1.. 3)

கீழேயுள்ள நிரலின் வெளியீடு என்னவாக இருக்கும் ?

```
# include <iostream.h>
# include <conio.h>

void main()
{
  clrscr();
  int i = 1, j = 1;
  while(i <= 3)
  {
        cout << ‘\n’;
        for(i=1;i<=j;i++)
          cout << ‘*’;
        i++;
        j++;
  }
  getch();
}
```

வெளியீடு ?

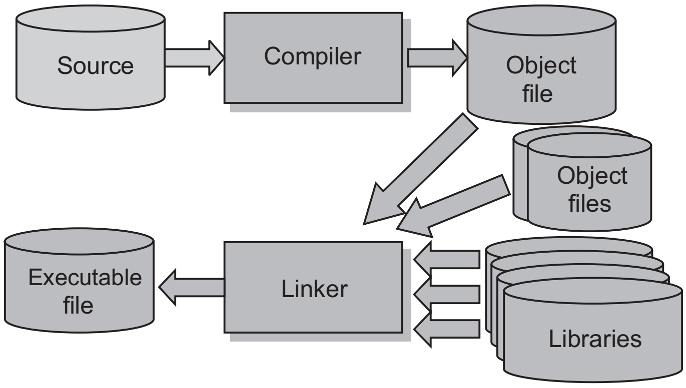
பின்னல் மடக்குகளை அமைப்பதற்கான விதிமுறைகள் :

1. வெளி மடக்கு, உள் மடக்கு இரண்டும் ஒரே கட்டுப்பாட்டு மாறியை வைத்துக் கொள்ள முடியாது. காரணம் அது தருக்கப் பிழைகளை ஏற்படுத்தும்.

2. உள்– மடக்கு முழுமையும் வெளி– மடக்கின் உடற்பகுதிக்குள் அமைந்திருக்க வேண்டும்.

12.6 நிரல் உருவாக்கம்

ஒரு கணிப்பொறி மொழியின் இலக்கணத்தைப் பயன்படுத்தி, உயர்நிலை மொழியில் நிரல்கள் எழுதப்படுகின்றன. உயர்நிலை மொழியில் எழுதப்படும் நிரல் மூலக் குறிமுறை (source code)எனப்படுகிறது. மூலக் குறிமுறையை எந்திரத்துக்குப் புரியும் வடிவில் மொழிமாற்றம் செய்ய வேண்டும். எந்திர மொழி வடிவில் மாற்றப்பட்ட நிரல் இலக்குக் கோப்பு (object file)எனப்படுகிறது. நிரல் பெயர்ப்பிகள் (compilers) மூலக்குறிமுறையிலிருந்து இலக்குக் கோப்புகளை உருவாக்குகின்றன. நிரல் பெயர்ப்பிகள் என்பவை, மூலக் குறிமுறையிலிருந்து எந்திர மொழிக்கு மாற்றம் செய்யும் மொழிபெயர்ப்பு நிரல்கள் ஆகும். நிரல்பெயர்ப்பி, மொழியின் இலக்கணத்தைச் (கட்டளை அமைப்பை) சரிபார்க்கிறது. பிழையில்லாத மூலக் குறிமுறையிலிருந்தே இலக்குக் கோப்பு உருவாக்கப்படுகிறது. இலக்குக் கோப்புடன் தேவையான நூலகக் கோப்புகள் தொடர்புறுத்தப்பட்டு, ஓர் இயக்குநிலைக் கோப்பு (executable file) உருவாக்கப்படுகிறது. செயல்பாடுகளின் இந்த வரிசை முறையைப் படம் 12.1ல் காண்க.



படம் 12.1 நிரல் செயலாக்கம்

பயிற்சி வினாக்கள்

1. கீழ்க்காணும் அறிவிப்புகளைச் சரியானது, பிழையானது எனக்குறிப்பிடுக. பிழையானது எனில், காரணம் கூறுக.

அறிவிப்பு	சரியானது/பிழையானது	காரணம்
int A;a;		
char name(10);		
float f,int;		
double d, float f;		
int 1choice, _2choice		

2. கீழ்க்காணும் நிரலில் பிழை திருத்தி, சரியான நிரலை எழுதுக

```
include <iostream.h>
include <conio.h>

void main()
{
  int N1,n1;
  cin << ' \nEnter two number s ..';
  result := N1 * n1;
  cout << ' \n' << Result;
}
```

3. கீழ்க்காணும் தகவல்களைப் பொருத்தமான அறிவிப்புக் கூற்றுகளாக எழுதிக் காட்டுக.

அ) 8/3 என்னும் கணக்கீட்டின் விடையை இருத்தி வைக்கவும்

ஆ) Emp_Name என்னும் மாறியில் "kalam" என்னும் தொடக்க மதிப்பிருத்துக.

இ) Y-yes, N-no எனக் குறிப்பிட்டுப் பயனரிடமிருந்து உள்ளீடு பெறுக.

4. கீழ்க்காணும் நிரல் பகுதிகளில் உள்ள பிழைகளைச் சுட்டிக் காட்டுக.

அ).
```
int a = 10; b = 5;
        if a > b
  cout << a;
```

ஆ).
```
if (a<b) && (a<0)
        cout << "a is negative and …"
```

இ).char option = ‘Y’;

```
do while option == ‘y’
{
cout << ‘*’;
……
}
```

ஈ).for(int i = 1; i < 10; i++)

```
cout << i * 2;
```

உ). do

```
{
cout << ‘*’;
} while(cout << “\nContinue <y/n>…”;cin>>ch;ch == ‘y’);
```

5. கீழ்க்காணும் நிரல்கள்/நிரல்பகுதிகளின் வெளியீடு என்னவாக இருக்கும் ?

```
// 5 a.
# include iostream.h>
# include <conio.h>

 void main()
 {
  int feet;
  const int inch_conversion = 12;
  clrscr();
  cout << “\nEnter feet …”;
  cin>>feet;
  cout << “\nConverted to inches …”
          << feet * inch_conversion;
  }
  input - 7 for feet
```

```
// 5 b.
# include <iostream.h>
# include <conio.h>
void main()
{
   int i = 1, sum = 0;
   clrscr();
   while(i++ <= 5)
  {
      cout << ‘\n’ << i;
      s += i;
  }
 cout << “\nValue of the variable i after
executing the while loop ..”
<< i << “\nSum :…”
<< s;
}
```

```
// 5 c
# include <iostream.h>
# include <conio.h>

void main()
{
  int i = 1, sum = 0;
  clrscr();
  while(++i <= 5)
  {
                cout << '\n' << i;
                sum += i;
  }
  cout << '\n' << i << '\t' << sum;
  getch();
}
```

```
// 5 d
# include <iostream.h>
# include <conio.h>
void main()
{
  int i = 1, sum = 0;
  clrscr();
  for(i = 1; i <= 5; i++)
  {           cout << '\n' << i;
                 sum += i;
  }
  cout << '\n' << i << '\t' << sum;
  for(i = 1; i <= 5; ++i)
  {
                 cout << '\n' << i;
                 sum += i;
  }
  cout << '\n' << i << '\t' << sum;
 }
```

```
//5e
# include <iostream.h>
# include <conio.h>

void main()
{
  int i = 1, j = 1;
  clrscr();
  do
  {
              while (j<=i)
              {
                 cout << '#';
                 j++;
              }
              cout << '\n';
              i++;
              j = 1;
  } while(i<= 5);
  getch();
}
```

```
// 5 f
# include <iostream.h>
# include <conio.h>

void main()
{
  int num = 1784, s= 0, d = 0, x;
  x = num;
  clrscr();
  for(;num > 0;)
  {          d = num % 10;
                s += d;
                num = num / 10;
  }
  cout << "\nThe sum of digits of "
       << x << "is : " << s;

  getch();
}
```

```
//5 g
# include <iostream.h>
# include <conio.h>
void main()
{
  clrscr();
  for(int i = 1,s = 0; ; i++)
  {
              if (i%2 == 0)
                 continue;
              s += i;
              if ( i > 9)
                 break;
  }
  cout << "\nThe sum is ...." <<
s;
  getch();
```

```
// 5 h
# include <iostream.h>
# include <conio.h>

void main()
{
  clrscr();
  for(int i = 1,x = 0;i <= 5; i++)
    x = x + i%2==0 ? i*1 : i * -1;
  cout << x;
  getch();
}
```

```
//5 j
# include <iostream.h>
# include <conio.h>

void main()
{
  clrscr();
  do
  {
    cout << "\ndo loop ...";
  } while (0);
  getch();
}
```

```
//5 k
# include <iostream.h>
# include <conio.h>

void main()
{
  clrscr();
  int i = 0;
  for(i = -5; i >= 5; i—)
              cout << "Bjarne Stroustrup";
  cout << "\nReturning to Edit Window..";
  getch();
}
```

```
//5 l
# include <iostream.h>
# include <conio.h>

void main()
{
  clrscr();
 int month = 5;
 if (month++ == 6)
    cout << "\nMay ...";
 else if (month == 6)
   cout << "\nJune ...";
 else if (—month == 5)
   cout << "\nMay again ..";

}
```

```
// 5 m
# include <iostream.h>
# include <conio.h>
void main()
{ int day = 3;
  switch (day)
  {
    case 0 : cout << "\nSunday ..";
    case 1 : cout << "\nMonday ..";
    case 2 : cout << "\nTuesday ..";
    case 3 : cout << "\nWednesday .. ";
    case 4 : cout << "\nThursday ..";
    case 5 : cout << "\nFriday ..";break;
    case 6 : cout << "\nSaturday ..";
  }
 }
```

```
// 5 n
# include <iostream.h>
# include <conio.h>

void main()
{
  clrscr();
  int bool = 2,b =4;
  while(bool)
  {
    cout << bool << '\t' << ++b << '\n';
    bool—;
    b—;
  }
  getch();

}
```

6. நிரல் எழுதுக.

அ) a மெய்யெண், b முழுஎண். a^b கண்டறியவும். (while() மடக்கு பயன்படுத்துக).

ஆ) கொடுக்கப்பட்ட எண்ணின் தொடர்பெருக்கல் (factorial) கண்டறியவும். (for() மடக்கு பயன்படுத்துக).

இ) n வரையுள்ள ஃபைபோனாசித் தொடரைக் கண்டறியவும். ஃபைபோனாசித் தொடர் 0,1,1,2,3,5,8,12,18,20,32,....

ஈ) கீழ்க்காணும் தோரணிகளை வெளியீடாகத் தருக. (தனித் தனி நிரல் எழுதுக.

```
1
1    2
1    2    3
1    2    3    4
1    2    3    4    5
```

```
4
3    4
2    3    4
1    2    3    4
```

```
A
B    C
D    E    F
G    H    I    J
```

7. ஒரு மாதத்தின் தேதியை உள்ளீடாகப் பெற்று. கீழ்காணும் செய்திகளை வெளியிடுக. switch() பயன்படுத்துக.

அ) day 1 எனில், "1st day in the month" என வெளியிடுக.

ஆ) day 2/22 எனில், "2nd /22nd day in the month" என வெளியிடுக.

இ) day 3/23, எனில், "3rd / 23rd day in the month" என வெளியிடுக.

ஈ) day 4,5.....14,15,16… எனில் 4th /5th .. day in the month" என வெளியிடுக.

பாடம் 13

செயற்கூறுகள் (Functions)

13.1 முன்னுரை

சி++ நிரல்களின் கட்டமைப்புக் கூறுகளாகத் திகழ்பவை செயற்கூறுகள் (Functions) ஆகும். ஒரு நிரலின் செயல்பாட்டுப் பகுதிகளாக அவை விளங்குகின்றன. ஒரு நிரலைச் செயல்படுத்துவதற்கான தொடக்கப் புள்ளி main() செயற்கூறாகும். செயற்கூறுகள்,

- ✓ நிரலின் நீளத்தைக் குறைக்கின்றன
- ✓ குறிமுறையின் மறுபயனாக்கத்துக்கு உதவுகின்றன

```
// To print X!/F! – Program - 13.1
# include < iostream.h >
# include <conio.h >
void main ()
{  int x, f,xfact=1, ffact = 1;
clrscr()
cout << "\nEnter values ...";
cin >> x >> f;   for (int a =1; a < = x;
a++)
xfact * = a;
for (a = 1; a< = f; a++)
ffact * = a;
cout << xfact/ffact;
getch();
}
```

```
// To print X!/F! using functions
// Program - 13.2
# include <iostream.h >
# include <conio.h >
int fact (int num)
{
int factorial = 1;
for (int a=1; a<= num; a++)
 factorial *= a;
return factorial;
}
void main ( )
{   int x, f;
   clrscr ( );
   cout<<"\nEnter values…";
   cin >> x >> f;
   cout << fact (x)/fact(f);
 }
```

Program 13.1 நிரலில் தொடர்பெருக்கலைக் கணக்கிடும் கட்டளைகள் திரும்பத் திரும்ப இரண்டுமுறை இடம் பெற்றுள்ளன. Program 13.2 நிரலில் தேவையானபோது fact(int num) என்னும் செயற்கூறு செயல்படுத்தப்பட்டுள்ளது. எனவே செயற்கூறுகள் கீழ்க்காணும் வகையில் உதவுகின்றன.

- ✓ நிரல் குறிமுறையின் மறுபயனாக்கம் (fact() செயற்கூறு இருமுறை அழைக்கப்பட்டுள்ளது).
- ✓ ஒரு செயற்கூறினை தனியாக நிரல்பெயர்த்து (compiling) வைத்துக் கொண்டால் ஒன்றுக்கு மேற்பட்ட நிரல்கள் அதனைப் பகிர்ந்து கொண்டு பயன்பெற முடியும்.

ஒரு செயற்கூறின் பல்வேறு பகுதிகளைக் காட்டும் அதன் பொதுவான கட்டமைப்பைக் காண்க.

```
// To print X!/F! using functions
// Program - 13.3
# include <iostream.h >
# include <conio.h >
int fact (int num)
{
  int factorial = 1;
  for (int a=1; a<= num; a++)
            factorial *= a;
  return factorial;
}

void main ( )
{   int x, f;
    clrscr ( );
    cout<<"\nEnter values…";
    cin >> x >> f;
    cout << fact (x)/fact(f);
}
```

அளபுருக்களுடன் கூடிய முன்வடிவு

return **கூற்று, செயற்கூறின் முடிவைக் சுட்டுகிறது. நிரலின் கட்டுப்பாட்டை செயற்கூறு அழைப்புக் கட்டளைக்கு அடுத்ததாக எடுத்துச் செல்கிறது.**

செயற்கூறினைச் செயல்படுத்தும் அழைப்புக் கூற்றுகள்.

13.2 செயற்கூறின் முன்வடிவு (Function Prototype)

செயற்கூறுகள், நிரலில் பயன்படுத்தப்படுவதற்கு முன்பாக அறிவிக்கப்பட வேண்டும். செயற்கூறு முன்வடிவின் மூலமாக ஒரு செயற்கூறு அறிவிக்கப்படுகிறது. எடுத்துக்காட்டாக Program **13.4.** நிரலைக் காண்க.

```
//Program -13.4
# include <iostream.h>
   void fun (char name [ ]) ;

   void main ( )
   {
     char n [ ] = { "C++ programming…."};
     fun (n);
   }
  void fun (char name[] )
  { cout << name; }
```

செயற்கூறு முன்வடிவு (செயற்கூறின் அறிவிப்பு)

செயற்கூறின் வரையறை

முன்வடிவு நிரல்பெயர்ப்பிக்குக் கீழ்க்காணும் தகவல்களை வழங்குகிறது.

1. செயலுருபுகளின் (arguments) எண்ணிக்கையும் அவற்றின் தரவினமும் (char name[ ]) என்பது ஒரு செயலுருபு).
2. திருப்பியனுப்பும் மதிப்பின் தரவினம் (மேற்கண்ட எடுத்துக்காட்டில் fun () என்னும் செயற்கூறின் தரவினம் void என்பதால், திருப்பியனுப்பும் மதிப்பு எதுவும் இல்லை என்பது பொருள். Program 13.2 ஐ மீண்டும் நோக்குக. fact() என்னும் செயற்கூறு திருப்பியனுப்பும் தரவினம் int ஆகும்.

செயற்கூறு முன்வடிவின் பொதுவான தொடர் அமைப்பு :

<தரவினம்> <செயற்கூறின் பெயர்> <செயலுருபுகள்>;

எடுத்துக்காட்டு:

```
void fun (char);
int max (int, int);
int max (int a, int b);
```

செயற்கூறு முன்வடிவின் முதன்மையான பயன்பாடு, செயற்கூறினுக்குத் தேவையான தரவுகளைச் சரிபார்த்துக் கொள்ள நிரல்பெயர்ப்பிக்கு உதவுவதே. ஒரு செயற்கூறினை அறிவிக்கும்போதும் வரையறுக்கும் போதும், முன் வடிவுடன் கூடிய ஒரு வார்ப்புரு (Template) பயன்படுத்தப்படுகிறது. ஒரு செயற்கூறு அழைக்கப்படும் போது, சரியான செயலுருபுகள் அனுப்பப்பட்டுள்ளனவா என்பதையும், திருப்பி அனுப்பும் மதிப்புச் சரியாக உள்ளதா என்பதையும் உறுதி செய்து கொள்ள நிரல் பெயர்ப்பி இந்த வார்ப்புருவைப் பயன்படுத்திக் கொள்கிறது. செயலுருபுகள் அல்லது திருப்பியனுப்பும் இனம் முன்வடிவில் உள்ளவற்றுடன் பொருந்தவில்லை எனில், நிரல்பெயர்ப்பின்போது, பிழையென நிரல் பெயர்ப்பி சுட்டிக் காட்டும்.

int max (int, int) **என்னும் முன்வடிவு சரியானதா ?**

ஒரு செயற்கூறு அறிவிப்பில், செயலுருபுகளின் பெயர்கள் மாதிரி மாறிகளே. எனவே அவற்றைக் குறிப்பிட வேண்டியது கட்டாயமில்லை. முன்வடிவில் குறிப்பிடப்படும் மாறிகள் வெறும் இடம் உணர்த்திகளாகவே (place holders) குறிப்பிடப்படுகின்றன.

செயற்கூறின் வரையறையில் செயலுருபுகளின் பெயர்கள் (மாறிகள்)தேவைப்படுகின்றன. காரணம், செயற்கூறினுள் செயலுருபுகள் எடுத்தாளப்படுகின்றன.

```
void fun (char name [ ] )
{
cout << name;
}
int fact (int num)
{
  int f = 1;
 for (int a = 1; a < = num; a ++)
```

செயலுருபுகள் name, num ஆகியவை செயற்கூறினுள் எடுத்தாளப்படுவது காண்க.

```
        f* = a;
    return f;
    }
```

13.3 ஒரு செயற்கூறினை அழைத்தல்

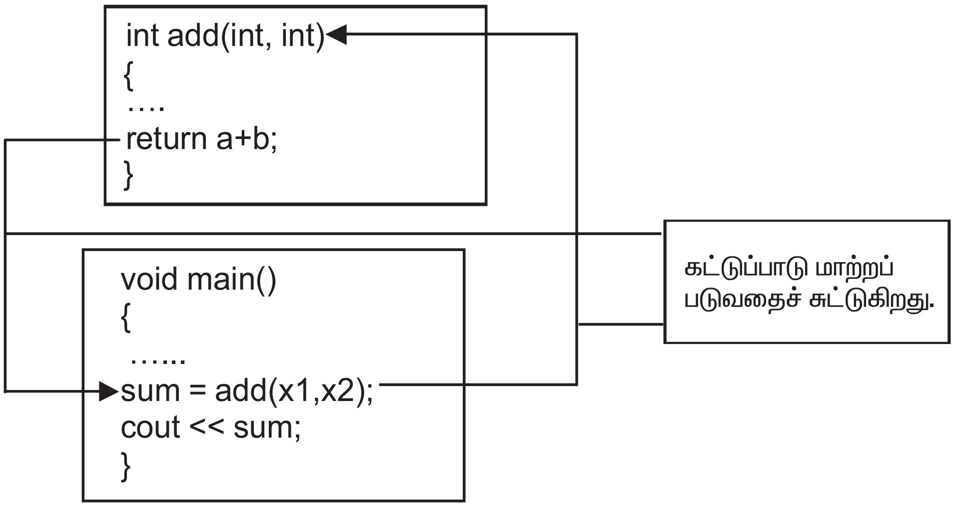
ஒரு செயற்கூறினை அதன் பெயர் குறிப்பிட்டு, வேறொரு செயற்கூறிலிருந்து அழைக்க முடியும். அதாவது, செயல்படுத்த முடியும். செயற்கூறின் பெயருடன் மெய்யான அளபுருக்கள் காற்புள்ளியால் பிரிக்கப்பட்டுப் பிறை அடைப்புக் குறிகளுக்குள் தரப்படும். எடுத்துக்காட்டு:

```
// Program - 13.5
# include <conio.h>
# include <iostream.h>

int add (int a, int b)

{ return a + b;}
void main ( )
{ int x1, x2, sum = 0
  cin >> x1 >> x2;

   sum = add (x1,x2);
  cout << sum;
}
```

முறையான அளபுருக்கள் (formal parameters) — int add (int a, int b)

மெய்யான அளபுருக்கள் (actual parameter) — sum = add (x1,x2);

ஒரு செயற்கூறின் செயல்பாடுகள் :



13.4 செயற்கூறினுக்கு அளபுருக்களை அனுப்பிவைத்தல்

செயற்கூறினை அழைக்கும் கூற்று, செயலுருபுகள் (Arguments)அல்லது அளபுருக்கள் (parameters) மூலமாக செயற்கூற்றோடு தகவல் பரிமாற்றம் செய்கிறது.

> அழைப்புக் கூற்றிலிருந்து செயற்கூறினுக்கும் செயற்கூறிலிருந்து அழைப்புக் கூற்றினுக்கும் இடையே தரவுகளின் பாய்வுக்கு அளபுருக்கள் வழித்தடங்கலாய் அமைகின்றன.

சி++ மொழியில் செயலுருபுகள் கொண்ட செயற்கூறுகளை, இருவகையில் செயல்படுத்த முடியும்.

- ✓ மதிப்பு மூலம் அழைத்தல் (Call by value)
- ✓ குறிப்பு மூலம் அழைத்தல் (Call by reference)

13.4.1 மதிப்பு மூலம் அழைத்தல் (Call by Value)

இந்தவகை அழைப்பின் போது அழைக்கப்பட்ட செயற்கூறு, தனக்கு அனுப்பி வைக்கப்படும் செயலுருபுகளின் மதிப்புகளை இருத்தி வைக்க, புதிய மாறிகளை உருவாக்குகிறது. மெய்யான அளபுருக்களின் (அழைப்புக் கூற்றில் குறிப்பிடப்படும் அளபுருக்கள்) மதிப்புகளை, முறையான அளபுருக்களில் (செயற்கூறின் தலைப்பில் குறிப்பிடப்படும் அளபுருக்கள்) நகலெடுக்கும். இவ்வாறு, செயற்கூறு தனக்கென அளபுருக்களின் நகலை உருவாக்கிப் பயன்படுத்திக் கொள்கிறது. எடுத்துக்காட்டு நிரல் Program 13.5ஐ நினைவு கூர்க.

```
// Program - 13.5
# include <iostream.h>
# include <conio.h>
int add (int a, int b)
{ return a+b;}
void main ()
{ int x1, x2, sum;
cin >> x1 >> x2;
sum = add (x, x2);
cout << sum;
}
Assume x1 = 5, x2 = 7
```

```
   Main()              add()

  x1 = 5              a = 5
  x2 = 7               b = 7
   sum =
Assume address of the variables :

  x1  = Oxf1          address   data
   x2 = Oxf3          Oxf1        5
   a   = Oxf7         Oxf2
   b   = Oxf9         Oxf3        7
 sum = Oxf6           Oxf4
                      Oxf5
                      Oxf6       12
                      Oxf7        5
                      Oxf8
                      Oxf9        7
```

மெய்யான அளபுருக்களான x1,x2 ஆகியவையும், முறையான அளபுருக்களான a, b ஆகியவையும் வெவ்வேறு நினைவக இருப்பிடங்களில் இருத்தப்பட்டுள்ளன என்பதைக் கவனித்தீர்களா ? ஆக, மதிப்பு மூலம் அழைக்கும் முறையில், எப்போதுமே, தரவுகள் அழைப்புக் கூற்றிலிருந்து, செயற்கூறின் வரையறைக்குப் பாயும்.

```
// Program - 13.6
// To exchange values
#include <iostream.h>
#include <conio.h>
# include <iomanip.h>
void swap (int n1, int n2)
{       int temp;
        temp = n1;
        n1 = n2;
        n2 = temp;
       cout << '\n'<<n1<<'\t'<<n2<<'\n';
}
```

```
void main ( )
{
  int m1 = 10, m2 = 20;
  clrscr ( );
  cout <<"\n Values before invoking swap" << m1 << '\t' << m2;
  cout << "\n Calling swap..";
  swap (m1, m2);
  cout << "\n Back to main.. Values are" << m1 << '\t' << m2;
  getch ( );
}
```

வெளியீடு

```
Values before invoking swap 10   20
Calling swap …..
20      10
Back to main…… Values are       10      20
```

m1, m2 ஆகிய மாறிகளின் மதிப்புகள் இட மாறியது, main() செயற்கூறில் பிரதிபலிக்கவில்லை. ஏன் என எண்ணிப் பார்த்தீர்களா **?**

செயலுருபுகள் மதிப்பு மூலமாக அனுப்பி வைக்கப்படும் போது, அழைக்கப்பட்ட செயற்கூறு, செயலுருபுகள் எந்தத் தரவினங்களைச் சார்ந்தவையோ அதே தரவினங்களில்

புதிய மாறிகளை உருவாக்குகிறது. செயலுருபுகளின் மதிப்புகள் புதியதாய் உருவாக்கப்பட்ட மாறிகளில் நகலெடுக்கப்படுகின்றன. எனவே, முறையான அளபுருக்களான இந்த மாறிகளில் செய்யப்படும் மாற்றங்கள், மெய்யான அளபுருக்களில் பிரதிபலிப்பதில்லை.

மதிப்பு மூலம் அழைத்தல் முறையில், முறையான அளபுருக்களில் செய்யப்படும் மாற்றங்கள் மெய்யான அளபுருக்களில் பிரதி பலிப்பதில்லை.

13.4.2 **குறிப்பு மூலம் அழைத்தல்** (Call by Reference)

இந்த முறையில், அழைக்கப்படும் செயற்கூறின் செயலுருபுகள் முறையான அளபுருக்கள்–அழைக்கும் செயற்கூறிலுள்ள மெய்யான அளபுருக்களின் மாற்றுப் பெயர்களாகச் (alias) செயல்படுகின்றன. செயற்கூறு தனது சொந்தச் செயலுருபுகளின் மீது செயல்படும் போது உண்மையில், மூலத் தரவுகளின் மீதே செயல்படுகிறது என்பது இதன் பொருளாகும். Program 13.6 ஐ நினைவு கூர்க. குறிப்புவகை அளபுருக்களைப் பயன்படுத்தி, swap () செயற்கூறினைத் திருத்தி எழுதுவோம்.

```
//Program - 13.7
// To exchange values
# include <iostream.h>
#include <conio.h>
void swap (int &n1, int &n2)
{
    int temp;
    temp = n1;
    n1 = n2;
    n2 = temp;
    cout<<'\n'<< n1
     <<'\t'<<n2<<'\n';
}
```

```
void main ( )
{
       int m1 = 10, m2 = 20;
       clrscr();
       cout<<"\nValues before swap call" << '\t' << m1 << '\t' << m2;
       swap(m1,m2);
       cout<<"\n Calling swap..";
       cout<<"\n Back to main.Values are"
       << '\t' << m1 << '\t'<< m2;
       getch ( );
}
```

வெளியீடு
Values before invoking swap 10 20
Calling swap..
20 10
Back to main. Values are 20 10

முறையான அளபுருக்களில் செய்த மாற்றங்கள் மெய்யான அளபுருக்களில் பிரதிபலித்துள்ளன. காரணம், முறையான அளபுருக்கள் குறிப்புவகை என்பதால் அவை மெய்யான அளபுருக்களின் நினைவக இடங்களையே சுட்டுகின்றன.

கீழேயுள்ள விளக்கங்களைக் காண்க.

main()	swap()
m1 = 10	n1 = 10
m2 = 20	n2 = 20
	temp

m1, m2 ஆகிய மாறிகளின் நினைவக முகவரிகள் முறையே Oxf1, Oxf4 என்க.

m1 = Oxf1 = 10

m2 = Oxf4 = 20

முறையான அளபுருக்களின் குறிப்பு (reference) என்பதை இவ்வாறு விளக்கலாம் :

m1 = 10;

n1 என்பது m1–ன் குறிப்பு என்பதை,

int &n1 = m1

எனக் குறிப்பிடலாம். இதன் பொருள், m1- மாற்றுப் பெயர் n1 என்பதாகும் அதாவது ,m1, n1 இரண்டுமே ஒரே நினைவக இருப்பிடத்தையே குறிக்கின்றன. எனவே, மேற்கண்ட கூற்றுகளின் பொருள்.

n1 = m1 = Oxf1 = 10

n2 = m2 = Oxf4 = 20

எனக் கொள்ளலாம்

முகவரி	**மாற்றத்துக்கு முன்**	**மாற்றத்துக்குப் பின்**
Oxf1 (n1, m1)	10	20
Oxf4 (n2, m2)	20	10

ஆக, முறையான அளபுருக்களில் செய்யப்படும் மாற்றங்கள் மெய்யான அளபுருக்களில் எதிரொலிக்கும்.

குறிப்பு மூலம் அழைத்தல் முறையில், முறையான அளபுருக்களில் செய்யப்படும் எந்த மாற்றமும் மெய்யான அளபுருக்களில் பிரதி பலிக்கும்.

கீழேயுள்ள நிரலை இயக்கிப் பார்க்கவும்.

```
// Reference variables
// Program -13.8
        # include <iostream.h>
        # include <conio.h>
        void main ( )
        {
          int num1 = 10, & num2 = num1;
          num2 ++;
          cout << num1;
}
```

இந்த நிரலின் வெளியீடு **11** எனக் கிடைக்கும்.

குறிப்பின் உதவியால் num1,num2 ஆகிய இரண்டு மாறிகளும் ஒரே நினைவக இருப்பிடத்தைச் சுட்டுமாறு செய்ய முடிகிறது. எனவே num1–ல் செய்யப்படும் மாற்றம் num2- ல் பிரதிபலிக்கிறது.

மெய்யான அளபுருக்களுக்கான விதிமுறைகள்

1. மதிப்பு வகையிலான முறையான அளபுருக்களுக்கு மாறிலி, மாறி அல்லது கோவையின் வடிவில் மெய்யான அளபுருக்களை அனுப்பிவைக்க முடியும்.

   எடுத்துக்காட்டாக,

   int add (int n1, int n2) என்ற முன்வடிவைக் கொண்ட ஒரு செயற்கூறினுக்கான அழைப்புக் கூற்று இவ்வாறு அமையலாம்.

   x = add (5, 10);

   x = add (a1, a2); [a1, a2 ஆகியவை int மாறிகள்]

2. குறிப்பு வகையிலான முறையான அளபுருக்களுக்கு, மெய்யான அளபுருக்களை மாறிகளின் வடிவிலேயே அனுப்பிவைக்க முடியும்.

   எடுத்துக்காட்டாக,

   int add (int & n1, int & n2) என்ற முன்வடிவு கொண்ட ஒரு செயற்கூறினுக்கான அழைப்புக்கூற்று.

x = add (a1, b1) (a1, b1 ஆகியவை int மாறிகள்) என்று அமைய வேண்டும்.

x = add ((a1 + b1, a1);

x = add (5,10); என்றெல்லாம் அமைவது பிழையாகும்.

```
// Program - 13.9
// To print 5 stars per row and 5 such rows
# include <iostream.h>
# include <conio.h>
void fun_starts (int &i)
{
      int j = 5;
      for (i= 1; i <= j; i++)
      cout << ‘ ‘<<‘*’;
}
void main ( )
{
       int mi = 1;
       clrscr( );
       for (; mi<= 5; mi++)
{
      cout << ‘\n’;
      fun_starts (mi);
}
   getch ( );
}
```

மேற்கண்ட நிரல், குறிப்புரையில் குறிப்பிட்டுள்ளபடி வெளியீட்டைத் தராத காரணம் என்னவென்று புரிகிறதா ? இந்த நிரல்,

* * * * *

என்று ஒரு வரி விடையை மட்டுமே தரும். ஐந்து வரி விடையைத் தராது.

காரணம் : **i -** என்னும் மாறி mi என்ற மாறியின் குறிப்பு (reference) ஆகும். முதல் முறை செயற்கூறு நிறைவேற்றப்படும் போது i–ன் மதிப்பு 6ஆகிவிடுகிறது. அதேநேரம் mi–ன் மதிப்பும் 6 ஆகிவிடும். எனவே main() ல் உள்ள for () மடக்கு ஒரேயொரு முறை மட்டுமே செயல்படுகிறது.

13.4.3 முன்னியல்புச் செயலுருபுகள் (Default parameters)

சி++ மொழியில் ஒரு செயற்கூறின் முன்வடிவில் முறையான அளபுருக்களில் முன்னியல்பு மதிப்புகளை இருத்திவைக்க முடியும்.

எடுத்துக்காட்டு

```
// Program - 13.10
// formal parameters with default values
# include <iostream.h>
# include <conio.h>
float power (float n, int p = 1)
{
   float prd = 1;
   for (int i = 1;      i<= p; i++)
   prd *= n;
   return prd;
}
void main ( )
{
    clrscr ( );
    int x = 4, b = 2;
    cout << “\n Call statement is power(b, x)...” << power (b,x);
    cout << “\n Call statement is power(b).. “ << power (b);
    getch ( );
}
```

வெளியீடு

Call statement is power (b, x) .. 16

Call statement is power (b) .. 2

power (b,x) என்னும் அழைப்புக் கூற்றில் தொடக்க மதிப்பிருத்தல் இவ்வாறு இருக்கும்.

n= b, p = x

power (b) என்னும் இரண்டாவது அழைப்புக் கூற்றில், n என்ற மாறியில் 2 என்னும் மதிப்பு இருத்தப்படும். மெய்யான அளபுரு இல்லையென்பதால் p என்னும் மாறியில் முன்னியல்பு மதிப்பான 1 இருக்கும்.

குறிப்பு:

- ✓ மாறியில் தொடக்க மதிப்பிருத்தும் வடிவில் முன்னியல்பு மதிப்பு தரப்பட்டுள்ளது.
- ✓ ஒரு செயற்கூறின் அழைப்புக் கூற்றில் சில அளபுருக்களை விட்டுவிடவோ அல்லது அளபுருக்கள் இல்லாமலே அழைக்கவோ 'முன்னியல்புச் செயலுருபுகள்' வழி வகுக்கின்றன.
- ✓ முன்னியல்பு மதிப்புகள் செயற்கூறு முன்வடிவில் செயலுருபுப் பட்டியலில் இறுதியாகவே இடம்பெற வேண்டும். பட்டியலின் தொடக்கத்திலோ, நடுவிலோ இடம் பெறக் கூடாது.

கீழேயுள்ள நிரலை இயக்கிப் பார்க்கவும்.

```
//Program - 13.11
# include <iostream h>
# include <conio.h>
int area (int side1 = 10, int side2=20)
{
   return (side1 * side 2);
 }
void main ( )
{
    int s1 = 4, s2 = 6;
    clrscr ( ) ;
    cout << area (s1, s2) << '\n';
    cout << area (s1) << '\n';
    cout << area (s2) << '\n';
    getch ( );
 }
```

```
வெளியீடு:
 24
 80
 120
மாறியில் மதிப்பிருத்தல்:
முதல் அமைப்பில்:
side1 = s1,side2=s2
2வது அமைப்பில்:
side1 = s1,side2=20
 3–வது அமைப்பில்:
side1 = s2,side2=20
```

கீழ்க்காணும் நிரலின் வெளியீடு என்னவாக இருக்கும் **?**

```
// Program - 13.12
// arguments with default values
# include <iostream.h>
# include <conio.h>
void print (int times, char ch = ' * ')
{
   cout << '\n';
   for (int i = 1; i < = times; i ++)
   cout << ch;
}
  void main ( )
{
   clrscr ( );
   print (50);
   print ('A', 97);
   print ( );
}
```

விடை:

print (50) - times என்னும் செயலுருபு 50 என்னும் மதிப்பை ஏற்கும். எனவே, 50 முறை* வெளியிடப்படும்.

print ('A', 97) - times என்னும் மாறியில் 'A' என்னும் மதிப்பு இருத்தப்படும் (char மதிப்பு int மதிப்பாக உள்ளுறை இனமாற்றம் நடைபெறும்). எனவே times-ல் 65 என்னும் மதிப்பு இருக்கும்.

97 என்னும் மதிப்பு ch - ல் இருத்தப்படும். இங்கே, int மதிப்பு char மதிப்பாக இனமாற்றம் பெறும். எனவே ch - ல் 'a' இருத்தப்படும்.

மெய்யான அளபுருக்கள், முறையான அளபுருக்களுடன் ஒன்றுக்கு ஒன்று என்ற அடிப்படையில் பொருத்தப்படும்.எனவே 'a' என்ற எழுத்து 65 முறை காட்டப்படும்.

print ( ) மெய்யான செயலுருபுகள் இல்லையென்பதால் முறையான அளபுருக்கள் முன்னியல்பு (default) மதிப்புகளை எடுத்துக் கொள்ளும். எனவே, * குறி, 50 முறை காட்டப்படும்.

13.5 மதிப்பினைத் திருப்பியனுப்புதல்

எந்த மதிப்பையும் திருப்பியனுப்பாத செயற்கூறு void என அறிவிக்கப்படுகிறது. ஒரு செயற்கூறினை அறிவிக்கும் போது அதன் தரவினம் வெளிப்படையாக குறிப்பிடப்படவில்லையெனில், அச்செயற்கூறின் இனம் int எனக்கொள்ளப்படும். எடுத்துக்காட்டாக,

```
int add (int, int);

add (int, int);
```

இரண்டு முன்வடிவுகளிலும் திருப்பியனுப்பும் மதிப்பு int ஆகும். காரணம் சி++ மொழியில் முன்னியல்பாக ஒரு செயற்கூறு திருப்பியனுப்பும் தரவினம் int ஆகும்.

கீழ்க்காணும் அட்டவணையைக் காண்க :

வரிசை எண்	செயற்கூறு முன்வடிவு	திருப்பியனுப்பும் இனம்
1	float power (float, int)	float
2	char choice ( )	char
3	char * success ( )	pointer to character
4	double fact (int)	double

13.5.1 குறிப்பு மூலம் திருப்பி அனுப்புதல்

குறிப்பு அல்லது மாற்றுப் பெயர் மாறிகள் :

```
// Program - 13.13
# include <iostream.h>
# include <conio.h>
  void main ( )
{        int i = 5;
         int &count = i ;
         cout << "\nCount: " << count;
         count ++;
         cout << "\ni: " << i;
         getch ( );
}
```

வெளியீடு

Count: 5

i: 6

count மாறி, **5** என்னும் மதிப்பை எப்படிப் பெற்றது **?**

count மாறியின் மதிப்பை மிகுக்கும் போது **i–**ன் மதிப்பு எவ்வாறு அதிகரித்தது **?** count மற்றும் i ஆகிய மாறிகள் நினைவகத்தில ஒரே தரவினையே சுட்டிக் கொண்டுள்ளன என்பதே இதற்குக் காரணம். இதேபோல, மூல மாறியில் ஏற்படும் மாற்றம் குறிப்பு மாறியில் எதிரொலிக்கும்.

இந்தக் கோட்பாட்டின் அடிப்படையில், கீழ்க்காணும் நிரலின் வெளியீடு என்னவாக இருக்கும் என்பதைக் கண்டறிக :

```
// Program - 13.14
# include <iostream h>
# include <conio.h>
int &maxref (int &a, int &b)
{      if (a>b)
       return a;
else
       return b;
}
void main ( )
{        int x = 20, y = 30, max = 0;
         max = maxref (x,y);
         cout << "\n Maximum is: " << max;
}
```

வெளியீடு

Maximum is : 30

மேற்கண்ட நிரலில், maxref() என்னும் செயற்கூறு ஓர் int இன மாறிக்குரிய குறிப்பினைத் (reference) திருப்பியனுப்புகிறது. maxref(x, y) என்னும் செயற்கூறு அழைப்பு, a, b, ஆகியவற்றில் எது பெரிதோ அதன் குறிப்பைத் திருப்பியனுப்பும். எனவே, max என்னும் மாறி y–ன் மதிப்பைப் பெறுகிறது.

கீழ்க்காணும் நிரலின் வெளியீடு என்னவாக இருக்கும் ?

```
// Program 13.15
# include <iostream h>
# include <conio.h>
int & maxref (int & a, int & b)
{ if (a > b)
return a;
        else
            return b;
}
void main ( )
{ int x = 20, y = 30, max = 0;
          maxref (x,y) = -1;
              cout << “\n Value of x is : “ << x;
                  cout << “\n Value of y is: “ <<y;
                   getch ( );
}
```

வெளியீடு :

Value of x is : 20

Value of y is : -1

குறிப்பு :

1. ஒரு குறிப்பினைத் திருப்பியனுப்பும் செயற்கூறின் அழைப்புக் கூற்று ஒரு மதிப்பிருத்தும் கூற்றின் இடப்பக்கம் இடம்பெற முடியும்.

2. மேற்கண்ட எடுத்துக்காட்டில், **y–**ல் **–1** இருத்தப்படும். காரணம், maxref() செயற்கூறு அழைப்பு, b என்னும் மாறிக்கான குறிப்பைத் திருப்பியனுப்பும் main() செயற்கூறில் bக்கு இணையான மாறி y ஆகும். எனவே, மதிப்பிருத்து கூற்றின் இடப்பக்கம் b இடம்பெறும். வலப்பக்கம் **–1** உள்ளது.

3. ஒரு குறிப்பு இனச் செயற்கூறின் முறையான அளபுருக்கள் எப்போதுமே குறிப்பு இன அளபுருக்களாகவே இருக்க வேண்டும். அதாவது,

```
int & maxref (int a, int b);
```

என வரையறுப்பின், நிரல்பெயர்ப்பில் பிழை கிட்டும். காரணம் a, b, ஆகிய மாறிகளின் செயல்பாட்டெல்லை maxref() செயற்கூறினுக்குள் மட்டுமே.

13.6 inline செயற்கூறுகள்

செயற்கூறுகளினால் ஏற்படும் நன்மைகளை எற்கெனவே பட்டியலிட்டுள்ளோம்.

✓ நிரல் குறிமுறையின் மறுபயனாக்கம், நினைவகத்தைச் சிக்கனமாகப் பயன்படுத்த உதவும்; நிரலின் நீளத்தைக் குறைக்கும்.

இத்தகைய நன்மைகள் ஒருபுறமிருக்க, ஒரு செயற்கூறினை அழைக்கும் கூற்று ? நிரல்பெயர்ப்பியை செயற்கூறு வரையறுக்கப்பட்டுள்ள இடத்துக்கு தாவச் செய்து, பிறகு அழைப்புக் கூற்றுக்கு அடுத்திருக்கும் ஆணைக்குத் தாவச் செய்கிறது. இது செவ்வனே நிறைவேற அடுக்கங்களை (stacks) உருவாக்கிக் கையாள வேண்டிய கூடுதல் வேலை நிரல்பெயர்ப்பிக்கு உள்ளது. செயல்கூறு அழைப்பு, திருப்பியனுப்பல், செயலுருபுகள் ஆகியவை தொடர்பான தனிச் சிறப்பு ஆணைகள் சிலவற்றைச் சேமித்து வைப்பதற்கு இத்தகைய அடுக்கங்கள் தேவைப்படுகின்றன. இதனால் நிரலின் செயல்படு வேகம் குறைகிறது. எனவே சில சூழ்நிலைகளில், குறிப்பாகச் செயற்கூறு சிறிதாக (குறைந்த எண்ணிக்கையிலான கட்டளைகள்) இருக்கும் போது, நிரல்பெயர்ப்பியானது செயற்கூறு அழைப்புக்கூற்று இருக்குமிடத்தில், அந்தச் செயற்கூறின் கட்டளைகளை நிரப்பிப் பதிலீடு செய்துவிடும். இந்த வசதி, **செயற்கூறினைப் பதிலிடல்** எனப்படுகிறது. அத்தகைய செயற்கூறுகள் inline செயற்கூறுகள் எனப்படுகின்றன.

inline செயற்கூறு மூல நிரலில் சாதாரணச் செயற்கூறு போன்றே தோற்றமளிக்கும். ஆனால் நிரல்பெயர்ப்பின்போது (compiling time) செயற்கூறின் கட்டளைகள் முழுமையும் அழைப்புக் கூற்றுக்குப் பதிலாக அப்படியே நிரலில் செருகப்பட்டுவிடும். inline செயற்கூறுகள் வேகமாகச் செயல்படும். ஆனால் அதிக நினைவக இடத்தை எடுத்துக் கொள்ளும்.

கீழேயுள்ள எடுத்துக்காட்டுகளைக் காண்க :

```
// Program - 13.16
// inline functions

# include <iostream.h>
# include <conio.h>

inline float convert_feet(int x)
{
  return x * 12;
}

void main()
{
  clrscr();
  int inches = 45;
  cout << convert_feet(inches);
  getch();
}
```

```
// working of Program - 13.16
// inline functions

# include <iostream.h>
# include <conio.h>


void main()
{
  clrscr();
  int inches = 45;
  cout << inches * 12 ;
  getch();
}
```

மேற்கண்ட எடுத்துக்காட்டில் உள்ளபடி, convert_feet(inches) என்னும் அழைப்புக்கூற்று, return கூற்றில் உள்ள (inches * 12) என்னும் கோவையால் பதிலீடு செய்யப்படும்.

ஒரு செயற்கூறினை inline செயற்கூறாக அறிவிக்க வேண்டும். எனில், program 13.16ல் காட்டியுள்ளபடி செயற்கூறின் தலைப்பில் inline என்னும் சிறப்புச் சொல்லை சேர்த்துக்கொள்ள வேண்டும்.

குறிப்பு : inline என்னும் சிறப்புச்சொல், நிரல்பெயர்ப்பிக்குத் தரப்படும் கோரிக்கை ஆகும். சில வேளைகளில் நிரல் பெயர்ப்பி, இந்தக் கோரிக்கையைப் புறக்கணித்துவிட்டுச் சாதாரண செயற்கூறாகவே கருதிக் கொள்ளும்.

13.7 மாறிகளின் வரையெல்லை (Scope) விதிமுறைகள்

'வரையெல்லை' என்பது ஒரு மாறியின் அணுகியல்பைக் (accessibility) குறிக்கிறது. சி++ மொழியில் நான்கு வகையான வரையெல்லைகள் உள்ளன. அவை :

1. உள்ளமை வரையெல்லை (Local scope)
2. செயல்கூறு வரையெல்லை (Function scope)
3. கோப்பு வரையெல்லை (File scope)
4. இனக்குழு வரையெல்லை (Class scope)

13.7.1 உள்ளமை வரையெல்லை (Local Scope)

```
// Program - 13.17
// to demonstrate local variable
# include < iostream.h
# include <conio.h>
void main ( )
{
     int a, b ;
     a = 10;
     b = 20;
if (a > b)
   { int temp; // local to this if block
     temp = a;
     a = b;
     b = temp;
}
   cout << '\n Descending order…';
   cout << '\n' <<a << '\n' <<b;
   getch ( );
}
```

- உள்ளமை மாறி, ஒரு தொகுதிக்குள் (Block) வரையறுக்கப்படுகிறது.
- ஓர் உள்ளமை மாறியின் வரையெல்லை அது வரையறுக்கப்பட்டுள்ள தொகுதிக்குள் மட்டுமே.
- ஓர் உள்ளமை மாறியை அது அறிவிக்கப்பட்டுள்ள தொகுதிக்கு வெளியிலிருந்து அணுக முடியாது.

program -13.18 உள்ளமை மாறியின் வரையெல்லையை விளக்குகிறது.

```
//Program - 13.18
# include <iostream.h>
# include <conio.h>
void main ( )
{    int a, b;
  a = 10
  b = 20;
  if (a > b)
 {     int temp;
     temp = a;
     a= b;
     b = temp;
 }
   cout << a << b << temp;
  getch ( );
}
```

இந்த நிரலை நிரல்பெயர்க்கும் போது, நிரல்பெயர்ப்பி ஒரு பிழை சுட்டும் செய்தியைக் காட்டும்.

Error in line no.13
The variable temp is not accessible.

உள்ளமை மாறிகளின் வாழ்நாள் அவை அமைந்துள்ள கட்டளைத் தொகுதி செயல்பட்டு முடியும்வரை மட்டுமே. அவற்றின் கட்டளைத் தொகுதிகள் செயல்பட்டு முடிந்தவுடன் உள்ளமை மாறிகள் அழிக்கப்படுகின்றன.

- உள்ளமை மாறிகள் அவை அறிவிக்கப்பட்டுள்ள கட்டளைத் தொகுதிக்கு வெளியே அறியப்படுவதில்லை. ஒரு கட்டளைத் தொகுதி என்பது நெளிவு அடைப்புக்குறிகளுக்குள் அமைந்திருக்கும்.
- உள்ளமை மாறிகள் அவை அறிவிக்கப்பட்டுள்ள கட்டளைத் தொகுதி செயல்படும் போது மட்டுமே நிலவும்.
- நிரலின் கட்டுப்பாடு ஒரு கட்டளைத் தொகுதிக்குள் நுழையும்போது, அதன் உள்ளமை மாறிகள் உருவாக்கப்படுகின்றன. வெளியேறும் போது அவை அழிக்கப்படுகின்றன.

program 13.19 நிரலில் உள்ளமை மாறிகளை அடையாளம் காட்டுக. அவற்றின் வரையெல்லையைக் குறிப்பிடுக.

```
// Program - 13.19
# include <iostream.h>
void main ( )
{ int flag = 1; a = 100;
  while (flag)
  {
    int x = 200;
   if (a > x)
  {  int j;
    -
   }
else
  { int h;
     -
  }}}
```

உள்ளமை மாறிகள்	வரையெல்லை
1. x	x: while() தொகுதிக்குள்
2. j	j:if(a>x) {} தொகுதிக்குள் மட்டும்
3. k	k:else{ } தொகுதிக்குள் மட்டும்

13.7.2. **செயற்கூறு வரையெல்லை** (Function Scope)

ஒரு செயற்கூறினுள் அறிவிக்கப்படும் மாறிகளின் வரையெல்லை, அந்த செயற்கூறின் கட்டளைத் தொகுதி மற்றும் அத்தொகுதிக்குள் அமைந்துள்ள உட்தொகுதிகள் அனைத்துக்கும் விரிகிறது.

program 13.19–ல் உள்ள flag என்னும் மாறியை main() செயற்கூறு முழுமையும் அணுக முடியும். அதனுள் இருக்கும் while(), if() ஆகிய உட்தொகுதிகளுக்குள்ளும் அணுக முடியும்.

செயற்கூறு வரையெல்லை கொண்ட ஒரு மாறியின் வாழ்நாள், அந்தச் செயற்கூறுத் தொகுதியின் வாழ்நாள் வரையாகும். ஒரு செயற்கூறின் முறையான அளபுருக்களின் வரையெல்லை செயற்கூறு வரையெல்லை கொண்டவை.

13.7.3. **கோப்பு வரையெல்லை** (File Scope)

அனைத்துக் கட்டளைத் தொகுதிகளுக்கும் செயற்கூறுகளுக்கும் மேலாக (குறிப்பாக main() செயற்கூறினுக்கு மேலே) அறிவிக்கப்படும் மாறி, கோப்பு வரையெல்லை கொண்டதாகும். கோப்பு வரையெல்லை கொண்ட மாறியின் வரையெல்லை அந்த நிரலின் முழுமையும் விரிகிறது. அதன் வாழ்நாள் அந்த நிரல் செயல்பட்டு முடியும் வரை நீடிக்கும்.

```
// Program - 13.20
// To demonstrate the scope of a variable
// declared at file level
# include <iostream.h>
# include <conio.h>
   int i = 10;
   void fun ( )
{ cout << i; }
void main ( )
{
            cout << i;
            while (i)
{
       ..................
       .................
             }
}
```

13.7.4 வரையெல்லை செயற்குறி (Scope Operator)

வரையெல்லை செயற்கூறி ஒரு மாறியின் மறைந்து கிடக்கும் வரையெல்லையை வெளிக் கொணரும். கீழேயுள்ள நிரலை நோக்குக.

```
// Program - 13.21
# include <iostream.h>
# include <conio.h>

int num = 15;

void main()
{
  clrscr();
  int num = 5;
  num = num + ::num;
  cout << num << '\t' << ++::num;
  getch();
}
```

num என்னும் மாறி main() க்கு உள்ளும், வெளியே கோப்பு வரையெல்லை கெண்டதாயும் அறிவிக்கப்பட்டுள்ளதைக் கவனித்தீர்களா? ::num எனக்குறிப்பிடப்பட்டுள்ளதை நோக்கினீர்களா? ::என்பது வரையெல்லை தீர்மானிப்புச் செயற்குறி எனப்படுகிறது. கோப்பு மட்டத்தில் அறிவிக்கப்பட்டுள்ள மாறிகளைக் குறிப்பிட இச்செயற்குறி பயன்படுகிறது. உள்ளமை மற்றும் கோப்பு வரையெல்லை கொண்ட மாறிகள் ஒரே பெயரில் நிலவும் சூழ்நிலைகளில் இச்செயற்குறி உதவும்.

13.7.5 இனக்குழு வரையெல்லை

பாடம் 15ல் இதுபற்றி விளக்கப்படும்.

பயிற்சி வினாக்கள்

1. கொடுக்கப்பட்ட விளக்கங்களின் அடிப்படையில் கீழ்க்காணும் செயற்கூறு முன்வடிவுகளைக் கட்டமைக்கவும்

அ) procedural-function ( )

-செயலுருபுகளை ஏற்காத, எந்த மதிப்பையும் திருப்பியனுப்பாத செயற்கூறு.

விடை :

void procedural - function (void);

அல்லது, void procedural - function ( ) ;

ஆ) manipulative - function ( )

-ஒரு double இனச் செயலுருபை ஏற்று, int மதிப்பைத் திருப்பியனுப்பும் செயற்கூறு.

விடை :

i. int manipulative - function (double); அல்லது

ii. int manipulative - function (double d); அல்லது

iii. manipulative - function (double)

இ) fun-default ( )

- இரண்டு செயலுருபுருகளை ஏற்கும். அவற்றுள் ஒன்று முன்னியல்பு int மதிப்பைக் கொண்டிருக்கும். மற்றது float மதிப்பாகும். எந்த மதிப்பையும் திருப்பியனுப்பாது.

விடை

void fun-default (float, int num = 10);

ஈ) return - reference - fun ( )

- இரண்டு int இனச் செயலுருபுகளை ஏற்று int இனக் குறிப்பைத் திருப்பியனுப்பும்.

விடை :

int & char - reference - fun (int &, int &);

உ) multi-arguments ( )

- இரண்டு float இனச் செயலுருபுகளை ஏற்கும். முதலாவது செயலுருபு pi-யின் மாறா மதிப்புடையது. இரண்டாவது செயலுருபு குறிப்பு இனத்தைச் சார்ந்தது எந்த மதிப்பையும் திருப்பியனுப்பாது.

விடை :

void multi - arguments (float const pi, float & a);

2. கீழ்க்காணும் செயற்கூறு முன்வடிவுகளில் பிழைகளைச் சுட்டுக :

அ. float average (a, b);

ஆ. float prd (int a,b);

இ. int default-arg (int a = 2, int b);

ஈ. int fun (int, int, double = 3.14);

உ. void strings (char [ ]);

3. கீழேயுள்ள செயற்கூறினை அழைப்பதற்குத் தேவையான அனைத்தும் கொண்ட main() செயற்கூறினை வரையறுத்துக் காட்டுக.

```
void line (int times, char ch)
{  cout << '\n';
   for (int i = 1; i < = times; i ++)
   cout << ch;
   cout << '\n';
}
```

4. இந்த நிரலில் பயன்படுத்தப்பட்டுள்ள மாறிகளின் வரையெல்லைகளைக் குறிப்பிடுக :

```
# include <iostream.h>              விடை :
float a, b ; void f1 (char);        a,b - கோப்பு வரையெல்லை
int main ( )                        ch -செயற்கூறு வரையெல்லை
{ char ch;                                   - main ( )
   ...........
{                                   i - அதன் தொகுதிக்குள் மட்டும்
   int i = 0;
   ............
   }
}
void f1 (char g)       x,y,g -செயற்கூறு வரையெல்லை - f1 ()
{
        short x, y ;
        ...............
   }
```

5. கீழ்காணும் நிரல்களிலுள்ள பிழைகளைச் சுட்டுக :

அ)

```
# include <iostream.h>

xyz (int m, int n)
{   int m = 10;
    n = m * n;
    return n;
}
void main( )
{ cout << xyz (9,27) ;}
```

விடை :

செயற்கூறின் தொகுதிக்குள் m என்னும் மாறியை அறிவிக்க இயலாது.

ஆ)

```
# include <iostream.h>

void xyz ( );
void main ( )

{ int x = xyz ( ) ; }
void xyz ( )
{ return '\0' ; }
```

விடை :

void இனமாக அறிவிக்கப்பட்டுள்ள செயற்கூறினுள் return கட்டளை இடம்பெறக் கூடாது. எனவே, செயற்கூறு அழைப்பு, ஒரு கோவையின் பகுதியாக இடம்பெற முடியாது.

இ)

```
# include <iostream.h>

void counter (int & a)
{ ++ a;}

void main ( )
{counter (50); }
```

விடை :

முறையான அளபுரு, குறிப்பு வகையாக இருப்பதால் மெய்யான அளபுருவை மதிப்பு வடிவில் அனுப்பி வைக்க இயலாது.

6. கீழ்காணும் நிரல்களின் வெளியீடு என்னவாக இருக்கும் **?**

அ)

```
# include <iostream.h>
int val = 10;
divide (int) ;
void main ( )
{
int val = 5;
val = divide (::val/val);
cout << :: val<<val;
}
  divide (int v)
{   return v/2;}
```

விடை: 101

ஆ)

```
# include <iostream.h>
divide (int v)
{ return v / 10;}
        void main ( )
{int val = -1;
val = divide (400) = = 40;
cout << "\n Val." << val;
}
```

விடை: 1

i) divide (400) என்ற அழைப்பு **40–ஐ** விடையாகத் தரும்

ii) divide (400) == 40என்பது **40**= =40 எனக் கொள்ளப்பட்டு, நிபந்தனை சரி என்பதால் val–ன் மதிப்பு **1** ஆகும்.

இ)

```
# include <iostream.h>
int incre (int a)
{ return a++; }
void main ( )
{int x = 10; x = incre (x);
cout << x;}
```

விடை: 10

ஈ)

```
# include <iostream.h>
# include <iostream.h>
void line( )
{static int v = 5;
int x = v - - ;
while (x)
{cout << ' * ' ; x — ;
}
cout << 'In';
}
void main ( )
{ clrscr ( );
for (int i = 1; i < = 5; i ++ )
line ( ) ;
getch ( );
}
```

<u>விடை :</u>

```
* * * * *
* * * *
* * *
* *
```

உ)

```
# include <iostream.h>
first (int i)
{ return i++; }
second (int x)
{ return x —; }
void main ( )
{ int val = 50;
val = val * val/val
val = second (val);
val = first (val);
cout << "\n Val: " << val;
}
```

<u>விடை :</u>

Val : 50

7. நிரல்கள் எழுதுக

அ) float cube (int,int,int) என ஒரு செயற்கூறினை வரையறுக்கவும். Cube()–ன் செயல்பாட்டைச் சோதிக்க main() செயற்கூறினை எழுதுக.

ஆ) unsigned long int factorial(int) என ஒரு செயற்கூறினை வரையறுக்கவும். ஓர் எண்ணின் தொடர்பெருக்கல் மதிப்பு அறியும் முறை: 5ன் தொடர்பெருக்கல் =1x2x3x4x5. factorial(n)- ன் மதிப்பை அறிய main() செயற்கூறினை எழுதுக.

இ) char odd-even-check(int) என்னும் செயற்கூறினை வரையறுக்கவும். இச்செயற்கூறு, தரப்படும் எண் இரட்டைப்படை எனில் ’E' எனவும், இல்லையேல் ’O' எனவும் விடை தரவேண்டும். இச்செயற்கூறினை இயக்கி, உரிய செய்தியை வெளியிட main () செயற்கூறினை எழுதிக் காட்டுக.

ஈ) int prime(int) என்னும் செயற்கூறினை வரையறுக்கவும். இச்செயற்கூறு, தரப்படும் எண் பகா எண் எனில் 1 எனவும், இல்லையேல் –1 எனவும் விடை தரவேண்டும். இச்செயற்கூறினை இயக்கி, உரிய செய்தியை வெளியிட main() செயற்கூறினை எழுதிக் காட்டுக.

பாடம் 14

கட்டமைப்புத் தரவினம்– அணிகள்

(Structured Data Type – Arrays)

14.1 முன்னுரை

சி++ மொழியில் **அணி** என்பது தருவிக்கப்பட்ட தரவினம் ஆகும். ஒரே தரவினத்தைச் சேர்ந்த பல மதிப்புகளை கொண்டிருக்கும்.

தரவுகள் அதிகமாக இருக்கும் போது, ஒரு தரவுத் தொகுதியில் ஒவ்வோர் உறுப்பாக அணுகிச் செயலாக்குவது கடினமான பணியாக இருக்கும். எடுத்துக்காட்டாக, கீழ்க்காணும் சூழ்நிலைகளை நோக்குக :

1. கொடுக்கப்பட்ட எண்களின் தொகுதியில் பெரிய எண்ணைக் கண்டறிதல்:

அ) தொகுதியில் இரண்டு எண்கள் இருப்பின், ஒப்பீடு இவ்வாறு இருக்கும்:

```
if (a > b)
        max = a;
else
        max = b;
```

ஆ) தொகுதியில் மூன்று எண்கள் இருப்பின், ஒப்பீடு இவ்வாறு இருக்கும்:

```
if (a > b) && (a > c)
        max = a;
else if (b > c)
        max = b;
else
        max = c;
```

இ) தொகுதியில் நான்கு எண்கள் இருப்பின், ஒப்பீடு இவ்வாறு இருக்கும்:

```
if (a > b && a > c && a > d)
        max = a;
else if (b > c && b > d)
        max = b;
else if (c > d)
        max = c;
else
        max = d;
```

எண்ணிக்கை அதிகம் ஆக ஆக ஒப்பீடுகளும் அதிகமாவதைக் கவனித்தீர்களா? தரவுகளைச் செயலாக்க மேற்கண்ட வழிமுறைகளைப் பின்பற்றினால், அதிகமான தரவுகளைக் கையாள்வது எளிதாக இருக்காது என்பதில் ஐயமில்லை.

இப்போது, கீழேயுள்ள கட்டளைத் தொகுதியை நோக்குங்கள்.

```
int a [4] = { 10, 40, 30, 20}; max = 0 ; i = 0;
        for (; i < 4; i ++)
        if a [i] > max
            max = a [i] ;
        cout << "\n The largest number is" << max;
```

மேற்கண்ட கட்டளைத் தொகுதி, கொடுக்கப்பட்ட எண்களின் பட்டியலில் பெரிய எண்ணைத் தீர்மானிக்கிறது. அதாவது, 10, 40, 30, 20, ஆகிய எண்களில் 40 என்ற எண்ணை வெளியிடுகிறது. if கூற்றை நோக்கினீர்களா? அதிகமான தரவுகளை எளிதாகக் கையாள, ஒரே தரவினத்தைச் சேர்ந்த உறுப்புகளின் தொகுதி ஓர் அணியாக அறிவிக்கப்பட்டுள்ளது.

ஆக, அணி என்பது ஒரே தரவினத்தைச் சேர்ந்த மாறிகளின் திரட்டு ஆகும். அணியின் உறுப்புகளை ஒரு பொதுப் பெயரால் குறிப்பிட இயலும்.

அணிகள் இருவகை உண்டு.

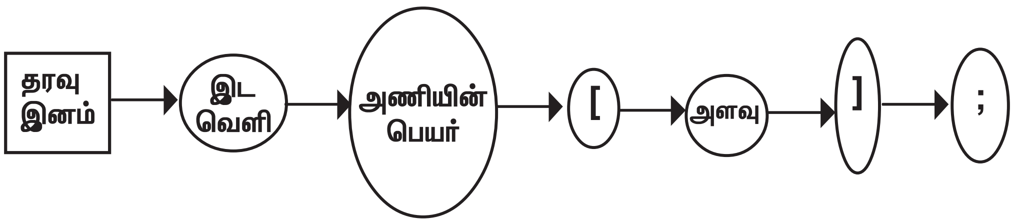
1. ஒரு பரிமாணம் : குறிப்பிட்ட எண்ணிக்கையில் ஒரே இன உறுப்புகளைக் கொண்டது.
2. பலபரிமாணம் : குறிப்பிட்ட எண்ணிக்கையில், ஒருபரிமாண அணிகளை உறுப்புகளாகக் கொண்டது.

14.2 ஒருபரிமாண அணி (Single Dimensional Array)

ஒரே இனத் தரவுகளின் பட்டியல்களைக் கையாள ஒருபரிமாண அணிகள் உகந்தவை ஆகும். ஒரு பரிமாண அணி இவ்வாறு அறிவிக்கப்படுகிறது:

```
int num_array [5];
```

கட்டளை அமைப்பு :



படம் 14.1 ஒருபரிமாண அணி

அணியின் உறுப்பெண்ணிக்கை எப்போதும் நேர்ம (positive) எண்ணாக இருக்கும்.

```
int num_array [5];
```

என்னும் அறிவிப்பின் பொருள், num_array என்பது ஐந்து முழு எண் மதிப்புகளைக் கொண்ட ஓர் ஒருபரிமாண அணி என்பதாகும். அணியின் ஒவ்வோர் உறுப்பையும் அணியின் பெயர், அணியில் உறுப்பின் இட இருப்பு (position)- இவற்றின் மூலம் அணுக முடியும். எடுத்துக்காட்டாக,

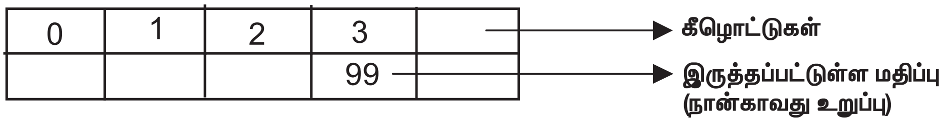
num-array [3] = 99,

என்னும் கட்டளை, அணியின் நான்காவது உறுப்பில் 99ஐ இருத்துகிறது. num_array என்பது அணியின் பெயர், [3] என்பது கீழொட்டு (Subscript) அல்லது இட இருப்பு (position) ஆகும்.

num_array- க்கு ஒதுக்கப்பட்டுள்ள நினைவக இடம் 10 பைட்டுகள் ஆகும். காரணம் அதில் ஐந்து int உறுப்புகள் உள்ளன. (ஓர் int மதிப்புக்கு 2 பைட்டுகள் தேவை. எனவே 5x2=10 பைட்டுகள் தேவை).

நினைவக ஒதுக்கீடு இவ்வாறு இருக்கும் :

num_array - அணியின் பெயர்



அணியின் கீழொட்டு எப்போதுமே 0–வில் தொடங்கும். எனவே num_array என்னும் அணி மாறியின் கீழொட்டுகள் 0 விலிருந்து 4 வரை இருக்கும். num_array[5] என்னும் கூற்றுப் பிழையானது, காரணம் ஏற்கத்தகு கீழொட்டுகள் 0 முதல் 4 வரை மட்டும். அணி அறிவிப்பின் சரியான பிற வடிவங்கள்

i. int array [100];

ii. float exponents [10];

iii. char name [30];

iv. const i = 10;
double val [i];

v. int days [ ] = {1,2,3,4,5,6,7};

iv) - வது எடுத்துக்காட்டில், val என்னும் அணியின் உறுப்பெண்ணிக்கை i என்னும் மாறிலி மூலம் குறிப்பிடப்பட்டுள்ளது. (v) வது எடுத்துக்காட்டில், days என்னும் அணியின் உறுப்பெண்ணிக்கை 7 என்பது மறைமுகமாக உணர்த்தப்பட்டுள்ளது. எவ்வாறென ஊகிக்க முடிகிறதா ? ஆம் {1, 2, 3, 4, 5, 6, 7} எனத் தொடக்க மதிப்பிருத்து கூற்றின் மூலம் தீர்மானிக்கப்படுகிறது. 7 உறுப்புகள் அணியின் அளவைத் தீர்மானிக்கிறது. இப்போது, program 14.1ஐ நோக்குங்கள். அணியில் நிகழ்த்தப்படும் அடிப்படைச் செயல்பாடுகளை அது விளக்குகிறது.

அணியைக் கையாள்வதற்கான எடுத்துக்காட்டுகள் :

cin >> number [4] - அணியின் ஐந்தாவது உறுப்புக்கு மதிப்பைப் பெற்றுக் கொள்கிறது.

cout << number [subscript]	-	subscript - இல் குறிப்பிடப்படும் உறுப்பினைத் திரையில் காட்டும்.
number [3] = number [2]	-	மூன்றாவது உறுப்பின் உள்ளடக்கத்தை அணியின் நான்காவது உறுப்பில் இருத்தும்.
number [3] ++	-	நான்காவது உறுப்பில் இருத்தப்பட்டுள்ள மதிப்பு ஒன்று மிகுக்கப்படும்.
number [++ a] = 10	-	++a– ஆல் குறிப்பிடப்படும் அணி உறுப்பில் 10 என்னும் மதிப்பு இருத்தப்படும்.

கீழேயுள்ள நிரல்களை இயக்கி வெளியீட்டை நோக்குக.

```
// Program - 14.1
// arrays and basic manipulations on
// arrays
# include <iostream.h>
# include <conio.h>
int a[5],ctr = 0, sum;
void main()
{
    for(;ctr<5;ctr++)
    {
    cout << "\nEnter value ..";
    cin >> a[ctr];
    }
  // generating sum of all the elements
  // stored in array
 for(ctr = 0, sum = 0; ctr<5;ctr++)
 sum+= a[ctr];
 clrscr();
 // display values stored in array
 for (ctr = 0; ctr < 5; ctr++)
 cout <<'\t' << a[ctr];
 cout << "\nSum of the elements ..." << sum;
 getch();
}
```

```
// try out
// Program – 14.2
# include <iostream.h>
# include <conio.h>
char ch [ ] = {'a', 'b', 'c', 'd', 'e', 'f'};
void main ()
{
      for (int i = 0; i < 6; i++)
          cout << ch[ i];
   for (j=5; j>=0; j—)
      cout << ch [j];
getch();
}
```

வெளியீடு

abcdeffedcba

```
// try out
// Program - 14.3
# include <iostream.h>
# include <conio.h>
void main ( )
{
int even [3] = {0, 2, 4}; int reverse [3];
for (int i=0, int j = 2; i<3; i++, j —)
      reverse [j] = even [i];
      clrscr ( );
for(i=0;i<3;i++)
       cout<< '\t' << even [i] << '\t' << reverse [i] << '\n';
      getch ( );
}
```

வெளியீடு

```
0      4
2      2
4      0
```

```
// try out
//Program -14.4
# include <iostream.h>
# include <conio.h>
void main ( )
{
   int x[5] = {1,2,3,4,5}, y [5] = {5,4,3,2,1},
   result [5] = { 0,0,0,0,0 };
   int i= 0;
   while (i++ < 5)
   result [i] = x [i] - y [i];
   clrscr ( );
   cout << "\n The contents of the array are: \n";
   i= 0 ;
  do
  {
      cout << '\t' << x [i]
           << '\t' << y [i]
           << '\t' <<  result [i]<<'\n';
      i++;
   } while (i<5);
   getch ( );
}
```

வெளியீடு

The contents of the array are:

1	-1	0
2	4	-2
3	3	0
4	2	2
5	1	4

```
//Try out
//Program - 14. 5
# include <iostream.h>
# include <conio.h>
void main ( )
{
        int vector [ ] = {2, 4, 8, 10, 0};
        for(int i=4; i>2; i—)
        vector [i]= vector [i-1];
        clrscr( );
        cout << "\n Elements of array before insertion \n";
        for (i= 0; i<5; i++)
        cout << vector[i];
        vector [2] = 6;
        cout << "\n Elements after insertion \n";
        for (i= 0; i<5; i++)
        cout << vector[i];
        getch ( );
}
```

வெளியீடு

```
Elements of array before insertion

248810

Elements after insertion

246810
```

ஓர் அணியிலுள்ள தரவுகளை ஏறுமுகமாக அல்லது இறங்குமுகமாக வரிசைப்படுத்த முடியும். இச்செயல்பாடு **வரிசையாக்கம்** (sorting) எனப்படுகிறது.

14.3 **சரங்கள்** (Strings)

சரங்கள், மதிப்புருக்கள் (literals) எனவும் அழைக்கப்படுகின்றன. அவை ஒரு பரிமாண char அணியாகக் கருதப்படுகின்றன. எண் வகை அணிகளை அறிவிப்பது போன்றே சரங்களை அறிவிக்க முடியும். எடுத்துக்காட்டாக,

i. char name [10];

ii. char vowels [ ] = {'a', 'e', 'i', 'o', 'u'};

iii. char rainbow [ ] = "VIBGYOR";

ஒரு char அணியை (ii), (iii) ஆகியவற்றில் உள்ளது போலத் தொடக்க மதிப்பிருத்தி அறிவிக்கலாம். (ii)–ல் உள்ளது போன்ற ஒரு char அணியைக் சரமாகக் கையாள வேண்டுமெனில் அதன் இறுதி உறுப்பாக '\0' (NULL) குறியுருவைக் குறிப்பிட வேண்டும்.

```
// Program - 14.6
// Reading values into an array of characters
# include <iostream.h>
# include <stdio.h>
# include <conio.h>
# include <string.h>
void main()
{
   clrscr();
   char name [30], address [30], pincode[7];
   cout << "\n Enter name ...";
   cin >> name;
   cout << "\n Enter address...";
   gets (address);
   cout << "\n Enter pincode ...";
   cin.getline (pincode, 7,'#');
   clrscr ( );
   cout << "\n Hello " << name;
   cout << "\n Your address is ..." << address;
   cout << "\n Pincode     ....";
   cout.write (pincode, sizeof(pincode));
   getch ( );
}
```

மேற்கண்ட எடுத்துக்காட்டில் (program-14.6), name, address, pincode ஆகிய மாறிகளுக்கான மதிப்புகள் cin, gets(), getline() ஆகியவற்றின் மூலம் பெறப்படுகின்றன. cin, வெற்று இடவெளி அல்லது நகர்த்தி திரும்பலை (Carriage Return-Enter key) சரத்தின் ஈற்றாக எடுத்துக் கொள்ளும். எடுத்துக்காட்டாக,

cin >> name;

அ) உள்ளீடாக K V N Perumal எனக் கொடுத்தால், name ல் K மட்டுமே இருத்தப்பட்டிருக்கும். காரணம் K–ஐ அடுத்து ஓர் இடவெளி.

ஆ) name ன் மதிப்பாக K.V.N. Perumal என உள்ளீடு தந்தால், name ல் K.V.N. Perumal என்ற முழுப் பெயரும் இருத்தப்படும்.

சர மதிப்பின் அங்கமாக இடவெளிகளையும் எடுத்துக் கொள்ள வேண்டுமெனில், stdio.h என்னும் கோப்பில் வரையறுக்கப்பட்டுள்ள gets() என்னும் செயல்கூறினைப் பயன்படுத்த வேண்டும். அல்லது அடிப்படை உள்ளீட்டுத் தாரை istream ன் உறுப்புச் செயற்கூறான getline() ஐப் பயன்படுத்த வேண்டும்.

கட்டளையமைப்பு :

gets ()	:	gets (char அணியின் பெயர்) அல்லது gets (char *)
getline()	:	cin.getline (char*, எழுத்துகளின் எண்ணிக்கை, வரம்புக்குறி);

சரத்தின் உள்ளடக்கத்தைத் திரையில் காட்ட இரண்டு வழிமுறைகள் உள்ளன.

1. cout << name; - பிற மாறிகளின் மதிப்பை வெளியிடுவதைப் போன்றதே
2. cout.write (pincode, 7); அல்லது cout.write (pincode, size of (pincode));

write ( ) என்பது அடிப்படை வெளியீட்டுத் தாரை அதாவது ostream-ன் உறுப்புச் செயற்கூறாகும். ஓர் இனக்குழுவின் அனைத்து உறுப்புச் செயற்கூறுகளையும், அந்த இனக்குழுவில் உருவாக்கப்பட்ட ஓர் இனப்பொருளின் (object/instance)மூலமாக அணுக வேண்டும். write() செயற்கூறினுக்குத் தேவையான இரண்டு அளபுருக்கள் சரத்தின் பெயர், காட்டப்பட வேண்டிய எழுத்துக்களின் எண்ணிக்கை. எடுத்துக்காட்டு:

```
//Program - 14.7
# include <iostream.h>
# include <conio.h>
void main()
{
      clrscr ( );
      char name[ ] = "Tendulkar";
      int i=1;
      while (i<10)
      {
         cout.write (name, i);
         cout << '\n';
         i++;
      }
     getch ( );
}
```

வெளியீடு

T

Te

Ten

Tend

Tendu

Tendul

Tendulk

Tendulka

Tendulkar

String.h கோப்பில் வரையறுக்கப்பட்டுள்ள, சரங்களைக் கையாளும் செயற்கூறுகளின் விளக்கத்தை அட்டவணை 14.1ல் காண்க.

வரிசை எண்	செயற்கூறு	கட்டளை அமைப்பு	பயன்பாடும் திருப்பியனுப்பும் மதிப்பும்
1	strlen ( )	strlen (char *)	சரத்திலுள்ள எழுத்துக்களின் எண்ணிக்கையைத் தரும். (எ.டு): char name [ ] = "Lasya"; strlen (name), 5 என்ற விடையைத் தரும்.
2	strcpy ( )	strcpy (char *,char *)	மூலச் சரத்தை இலக்குச் சரத்தில் படியெடுக்கும் (எ.டு.) strcpy (name, petname)
3	strcmp ( )	strcmp (char*, char*)	இரண்டு சரங்களை ஒப்பிட்டு, இரண்டு சரங்களும் நிகர் எனில், 0 என்ற விடையைத் தரும். அகர வரிசைப்படி முதல்சரம் இரண்டாவது சரத்துக்கு முன்பு இருப்பின் 1 என்னும் விடையைத் தரும். பின்பு இருப்பின் –1 என்ற விடை கிடைக்கும். (எ.டு.) strcmp (“Abc”, “Abc”) என்பது 0 என்னும் விடை தரும். strcmp (“Abc”, “abc”) என்பது –1 என்னும் விடைதரும். strcmp ("abc", "Abc") என்பது 1 என்னும் விடை தரும்.

அட்டவணை 14.1 சரச் செயற்கூறுகள்

சரங்களை ஓர் ஒருபரிமாண அணி போலவே ஒவ்வொர் எழுத்தாக எடுத்தாள முடியும்.

எடுத்துக்காட்டு:

```
// Program - 14.8
# include <iostream.h>
# include <conio.h>
void main ( )
{
    clrscr();
    char name[ ] = “Pascal”, reverse[7];
    int i= 0;
    while (name[i] != ‘\0’)
    i++;
    reverse[i] = ‘\0’;
    —i;
    int j = 0;
    while (i>= 0)
    reverse [i—] = name [j++];
    cout << “\n The contents of the string are: “<< name;
    cout << “\n The contents of the reversed string ...” << reverse;
getch ( );
}
```

கீழ்க்காணும் நிரலின் வெளியீடு என்னவாக இருக்கும் ?

```
//Program - 14.9
# include <iostream.h>
# include <conio.h>
# include <string.h>
main()
{
   char word [ ] = “test”;
   int i=0, k = 4, j = 0;
   clrscr( );
   while (i < 4)
  {
     j = 0;
    while (j<=i)
        cout << word [j++];
    cout << ‘\n’;
    i++;
  }
  return 0;
}
```

14.4 இருபரிமாண அணி (Two-Dimensional Array)

இருபரிமாண அணி என்பது, ஒரு பரிமாண அணிகளை உறுப்புகளாகக் கொண்ட ஓர் அணியாகும். எடுத்துக்காட்டாக, marks[3] [4] என்னும் அணி, 3 கிடக்கைகள், 4 நெடுக்கைகள் கொண்ட ஒரு அட்டவணையைக் குறிக்கும்.

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3

```
int makrs [3] [4] = {90, 70, 80, 0, 75, 95, 65, 0, 80, 90, 90, 0};
```

என்னும் இருபரிமாண அணி,

	0	1	2	3
0	90	70	80	0
1	75	95	65	0
2	80	90	90	0

என்னும் அட்டவணையை உருவாக்கும்.

குறிப்பு :

√ ஓர் இருபரிமாண அணியின் உறுப்பெண்ணிக்கை, கிடக்கை நெடுக்கை எண்ணிக்கைகளின் பெருக்குத் தொகையாகும். மேலே கண்ட எடுத்துக்காட்டில் marks என்னும் அணி 12 (3 x 4) உறுப்புகளைக் கொண்டுள்ளது.

√ அணி உறுப்புகளைக குறிக்கும் கீழொட்டுகள் (subscripts) எப்போதுமே 0–வில் தொடங்குகின்றன. கிடக்கைகளைக் குறிக்கும். கீழோட்டு 0 முதல் 2 வரையும், நெடுக்கைகளுக்கு 0 முதல் 3 வரையும் இருக்கும்.

√ இருபரிமாண அணியின் உறுப்பு [கிடக்கை] [நெடுக்கை] எனக் குறிப்பிடப்படும். எடுத்துக்காட்டாக,

```
marks [0] [3] = marks [0] [0] + marks [0] [1] + marks [0] [2];
```

என்னும் கட்டளை, முதல் கிடக்கையிலுள்ள மதிப்பெண்களை, அதாவது 90, 70, 80 ஆகியவற்றின் கூட்டுத் தொகையைக் கணக்கிடும். ஓர் இருபரிமாண அணி இவ்வாறு அறிவிக்கப்படுகிறது:

< தரவினம்> <அணிப்பெயர்> [<கிடக்கை>] [<நெடுக்கை>];

எடுத்துக்காட்டு:

1. int a[3] [2]- a என்னும் அணி 3 கிடக்கைகள், 2 நெடுக்கைகள் கொண்டது.
2. const int i=5;

float num [i] [3]; - num என்னும் இரு பரிமாண அட்டவணை **5** கிடக்கைகள், **3** நெடுக்கைகள் கொண்டது.

3. short fine [‘A] [‘E’] - fine என்னும் இரு பரிமாண அட்டவணை **65** கிடக்கைகள், **69** நெடுக்கைகள் கொண்டது.

குறிப்பு :

அணியின் பரிமாணங்கள் (கிடக்கைகள்/நெடுக்கைகள்) இவ்வாறெல்லாம் இருக்க முடியும்.

1. int மாறிலிகள்
2. முழு எண் அல்லது வரிசையெண் சார்ந்த மாறிலிக் குறிப்பெயர்கள்
3. char மாறிலிகள்
4. enum குறிப்பெயர்கள்

14.4.1 இருபரிமாண அணிகளின் நினைவக இருப்பிடம்

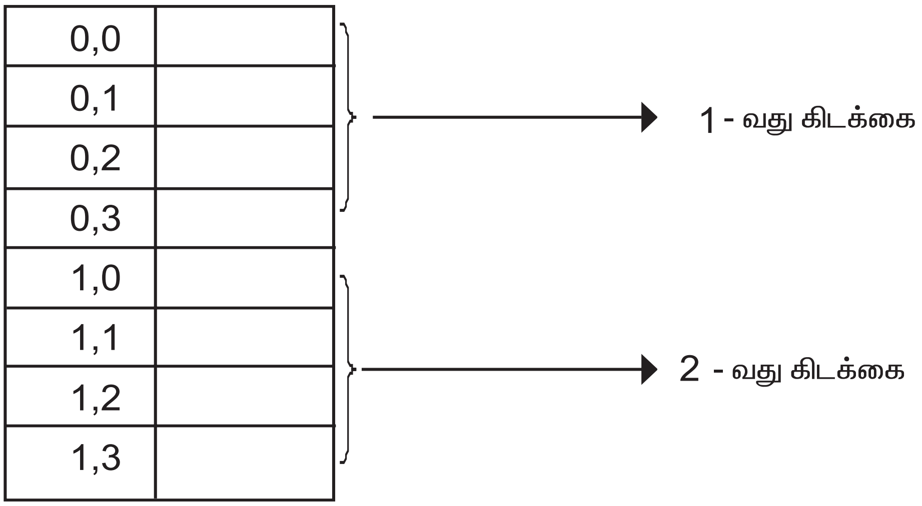
ஓர் இருபரிமாண அணியின் உறுப்புகள் நினைவகத்தில் தொடர்ச்சியான இருப்பிடங்களில் இருத்திவைக்கப்படுகின்றன. உறுப்புகளின் மதிப்புகள் கீழ்க்காணும் முறைகள் ஒன்றில் இருத்தப்படுகின்றன.

1. கிடக்கை வாரியாக: கிடக்கை முதன்மை (row-major) என்று பெயர்
2. நெடுக்கை வாரியாக: நெடுக்கை முதன்மை (column-major) என்று பெயர்

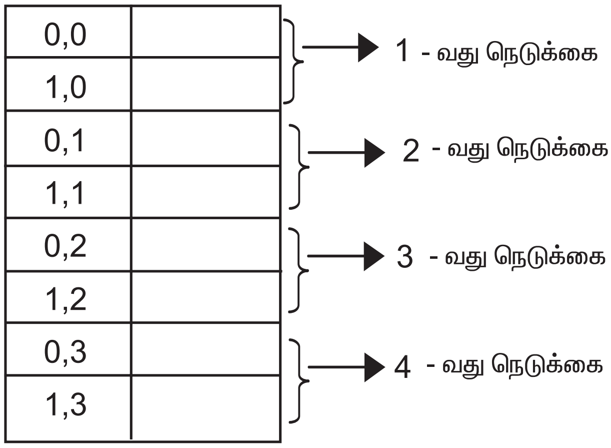
எடுத்துக்காட்டு

int sales [2] [4] என்ற அணியின் உறுப்புகள் இவ்வாறு அமைகின்றன.

கிடக்கை – முதன்மை வரிசை



நெடுக்கை– முதன்மை வரிசை



sales[1][0] என்ற உறுப்பின் இட இருப்பைக் கிடக்கை முதன்மை வரிசையிலும், நெடுக்கை முதன்மை வரிசையிலும் கவனித்தீர்களா ?

இருபரிமாண அணியின் நினைவகக் கொள்ளவு இவ்வாறு கணக்கிடப்படுகிறது.

உறுப்பெண்ணிக்கை X ஓர் உறுப்புக்குத் தேவையான நினைவக அளவு.

எடுத்துக்காட்டாக, int sales[2][4] என்ற அணியின் அளவு இவ்வாறு கணக்கிடட்பபடுகிறது.

உறுப்பெண்ணிக்கை = கிடக்கைகள் X நெடுக்கைகள் = 2 x 4 = 8

∴ 8 x 2 (ஓர் int மதிப்புக்கு 2 பைட்டுகள்)

∴ அளவு = 16 பைட்டுகள்

float num [4][6] என்ற அணியின் அளவு என்னவாக இருக்கும் ?

விடை :

4 x 6 x 4 = 96 பைட்டுகள்

கீழ்க்காணும் அணியை நோக்குக.

int num [4] [3] = {78, 7, 6, 4, 5, 8, 9, 7, 6, 1, 2, 3};

num		0	1	2
	0	8	⑦	6
	1	4	⑤	8
	2	⑨	7	6
	3	1	②	3

num அட்டவணையில் வட்டமிடப்பட்ட மதிப்புகளின் உறுப்புகளை எழுதிக் காட்டுக:

விடை :

num [0] [1] - (7)

num [1] [1] - (5)

num [2] [0] - (9)

num [3] [1] - (2)

கீழே அறிவிக்கப்பட்டுள்ள அணிகளில், உறுப்புகளின் எண்ணிக்கையைத் தீர்மானிக்கவும் :

a) int array [10] [12]; விடை: **120 உறுப்புகள்**

b) int x [ ] [2] = {0,1,1,2,2,3} விடை: **6 உறுப்புகள் (கிடக்கை –3, நெடுக்கை–2)**

தொடக்க மதிப்புகளோடு அணியை அறிவிக்கும் போது, முதல் பரிமாணத்தைக் குறிப்பிடுவது கட்டாயமில்லை.

```
        #include <iostream.h>
        # include <conio.h>
        #include <iomanip.h>
        void accept (int s [3] [4], int total)
        { int r = 0;  c = 0;
                for (; r < 3; r ++)
                {cout << "\n Month: " << r+1;
                   for (; c < 4; c++)
                      {cout << '\n' << c+1 <<
"Quarter..";
```

செயல்கூறுகளுக்கு, அணிகளைச் செயலுருபுகளாக அனுப்பிவைக்க முடியும். அணியின் பெயரை மட்டும் மெய்யான அளபுருவாகக் குறிப்பிட்டால் போதும். பரிமாணங்களைத் குறிப்பிடத் தேவையில்லை.

அணி அளபுருக்கள் இயல்பாகவே குறிப்புவகை அளபுருக்கள் போலச் செயல்படுகின்றன. காரணம், அணியின் பெயர், நினைவகத்தில அவ்வணி இருத்தப்பட்டுள்ள பகுதியின் தொடக்க முகவரியைக் குறிக்கிறது. பிற மாறிகளின் பெயர்கள் அவ்வாறில்லை. எனவே அணியின் பெயரை செயலுருபாகக் குறிப்பிடுவது, ஒரு முகவரியை முறையான அளபுருவுக்கு அனுப்பி வைக்கிறோம் என்று பொருள் (குறிப்புவகை அளபுருவைப் போல)

```
// Program - 14.10
#include <iostream.h>
# include <conio.h>
void accept (int s[3][4], int &total)
{
   int r = 0, c = 0;
   for (; r < 3; r++)
{
   cout << "\n Month: " << r+1;
   for (c = 0; c < 4; c++)
   {
           cout << '\n' << c+1 << " Quarter..";
           cin >> s[r][c];
           total += s[r][c];
           }
     }
}
void display (int d[3][4], int total)
{
      int r, c;
      clrscr ( );
      cout << "\nSales figures for 3 months & their respective quarters..";
      for (r = 0; r < 3; r++)
     {
          cout << "\n Month ..." << r+1;
           for (c = 0; c < 4; c ++)
 cout << '\t' << d[r][c];
     }
      cout << "\n Total sales .." << total;
}
void main ( )
{
           clrscr();
           int sales[3][4], t = 0;
           accept(sales,t);
           display(sales,t);
            getch();
}
```

இப்போது கீழேயுள்ள நிரலை நோக்குக:

```
// Program - 14.11
# include <iostream.h>
# include <conio.h>
void accept (int a)
{
        cout << "\n Enter a number ..";
        cin >> a;
}
void display (int a)
{
        cout << '\n' << a;
}
void main ( )
{
        int num [2][2] ={{0,0},{0,0}}, r = 0, c = 0;
        clrscr ( );
        for (; r < 2; r++)
        for (; c < 2; c++)
        accept (num[r][c]);
        clrscr();
        for (r = 0; r < 2; r++ )
        for (c = 0; c < 2; c++)
        display (num[r][c]);
       getch();
}
```

accept() செயல்கூறினுக்கு உள்ளீடாக 1, 2,3, 4, ஆகிய மதிப்புகள் தரப்படுவதாகச் கொள்க. விடை:

0

0

0

0

எனக் கிடைக்கும். num அணியின் உறுப்புகளில் 1, 2, 3, 4, ஆகிய மதிப்புகள் ஏன் இருத்தப்பட வில்லை என எண்ணிப் பார்த்தீர்களா ?

இந்த எடுத்துக்காட்டில், void accept() செயல்கூறினுக்கு அளபுருவாக தனித்தனி உறுப்புகள் அனுப்பிவைக்கப்படுகின்றன. எனவே, அவை மதிப்பு வகை அளபுருவாகவே எடுத்துக் கொள்ளப்படும். குறிப்புவகை அளபுருவாகக் கருதி கொள்ளப்படாது.

குறிப்பு:

அணியின் பெயர் மட்டுமே அணியின் தொடக்க முகவரியைக் குறிக்கிறது.

இப்போது, மேற்கண்ட நிரலில் accept() செயல்கூறினைச் சற்றே மாற்றி எழுத வேண்டும். 1, 2, 3, 4, என அதே உள்ளீடுகளைத் தந்து, வெளியீடாக,

1

2

3

4

எனப் பெறமுடியும்

14.4.2 அணிக் கோவை (Matrix)

அணிக்கோவை என்பது m x n எண்களை m கிடக்கைகளிலும் n நெடுக்கைகளிலும் ஒரு செவ்வக அணி வடிவில் கொண்டிருக்கும். அணிக்கோவைகளை இருபரிமாண அணிகளாக வரையறுக்க முடியும்.

Program 14.12 நிரல் இரண்டு அணிக்கோவைகளுக்கான மதிப்புகளை உள்ளீடாகப் பெற்று அவ்வணிக் கோவைகளின் நிகர்ப்பாட்டை (equality) பரிசோதித்து விடை தருகிறது.

14.5 சரங்களின் அணி

சரங்களின் அணி என்பது ஓர் இருபரிமாண char அணி ஆகும். அணி வரையறுப்பில் உள்ள முதல் சுட்டெண் (கிடக்கை எண்ணிக்கை) சரங்களின் எண்ணிக்கையைக் குறிக்கும். இரண்டாவது சுட்டெண் (நெடுக்கை எண்ணிக்கை) சரங்களின் உச்ச அளவு நீளத்தைக் குறிக்கும். எடுத்துக்காட்டு:

```
char day-names [7] [10]    =    {“Sunday”,
                                 “Monday”,
                                 “Tuesday”,
```

“Wednesday”,

“Thursday”,

“Friday”,

“Saturday”};

இந்த அணியின் உறுப்புகளிலுள்ள மதிப்புகள் அட்டவணை **14.1** இல் உள்ளவாறு இருத்தப்பட்டிருக்கும்.

```
//Program - 14.12
# include <iostream.h>
# include <conio.h>
void accept (int mat[3][3])
{
    clrscr();
    int r = 0, c = 0;
    for (; r < 3; r++)
   {
        cout << “\n Enter elements for row..” << r;
        for (c=0; c < 3; c++)
        cin >> mat[r][c];
    }
}
void main ( )
{
         int m1[3][3], m2[3][3];
         accept (m1);
         accept (m2);
         int i=0, j = 0, flag = 1;
         for (; i < 3; i++)
{
         for (; j < 3; j ++)
         if (m1[i][j] != m2[i][j])
         {
                flag = 0;
                break;
         }
                 if (flag == 0)
                 break;
         }
         if (flag)
         cout << “\n The matrices are equal ...”;
else
         cout << “\n The matrices are not equal..”;
         getch ( );
}
```

	0	1	2	3	4	5	6	7	8	9	
0	S	u	n	d	a	y	\0				day-names [0]
1	M	o	n	d	a	y	\0				day-names [1]
2	T	u	e	s	d	a	y	\0			day-names [2]
3	W	e	d	n	e	s	d	a	y	\0	day-names [3]
4	T	h	u	r	s	d	a	y	\0		day-names [4]
5	F	r	i	d	a	y	\0				day-names [5]
6	S	a	t	u	r	d	a	y	\0		day-names [6]

அட்டவணை 14.1 நினைவகத்தில் அணியின் உறுப்புகள்

அணியிலுள்ள தனித்தனிச் சரங்களைக் கையாள day-names[0] என முதல் சுட்டெண் (கிடக்கை எண்) மட்டும் குறிப்பிட்டால் போதும். ஒரு சரத்திலுள்ள ஒரு குறிப்பிட்ட எழுத்தை அதாவது அணியின் ஒரு தனி உறுப்பைக் கையாள day-names[0][5] என இரண்டு சுட்டெண்களையும் (கிடக்கை / நெடுக்கை எண்களை) குறிப்பிட வேண்டும்.

ஒவ்வொரு சரத்தின் இறுதியிலும் வெற்றுக் குறியுருவைச் (null character-'\0') சேர்ப்பது கட்டாயமில்லை. (optional). நாம் சேர்க்காவிட்டாலும் சி++ நிரல்பெயர்ப்பி தானாகவே இணைத்துக் கொள்ளும்.

பயிற்சி வினாக்கள்

1. கீழ்க்காணும் கட்டளைகள் பிழைசுட்டுவது ஏன் ?

அ) int a [5.5];

அணியின் பரிமாணம் முழு எண்ணாகவே இருக்க வேண்டும்.

ஆ) float f [3] = {1.0, 2.0};

- இது பிழை சுட்டாது. ஆனால் அணியில் குறிப்பிடப்பட்டுள்ள உறுப்பு எண்ணிக்கையை விடக் குறைவான மதிப்புகளே தரப்பட்டுள்ளன. இவ்வாறு குறைவாக இருக்குமெனில், மீதி உறுப்புகளில் 0 மதிப்பு தாமாகவே இருத்தப்பட்டுவிடும்.

இ) float num [A];

அணியின் பரிமாணம் வெளிப்படையாக அறிவிக்கப்பட வேண்டும். இதில், A என்னும் மாறிக்கு மதிப்பு எதுவும் கிடையாது. இக்கூற்றையே,

float num [‘A’] அல்லது

const A = 10; float num [A];

என மாற்றி எழுதலாம்.

ஈ) char a [3] [ ] = {“one”, “two”, “three”};

– தொடக்க மதிப்புடன் அறிவிக்கப்படும் ஓர் இருபரிமாண அணியில் முதல் பரிமாணத்தை விட்டு விடலாம். இரண்டாவது பரிமாணம் கட்டாயமாக இடம் பெற வேண்டும். எனவே மேற்கண்ட கூற்றை,

char a [ ] [6] = {“one”, “two”, “three”};

என மாற்றி எழுதலாம்.

உ) char ch [1] = ‘s’;

-Char அணியை சரமாக அறிவிக்க வேண்டுமெனில்,

char ch [2] = “s”

என அறிவிக்கலாம். அல்லது char அணியாகவே அறிவிக்க வேண்டுமெனில்,
char ch [1] = {‘s’}

என அறிவிக்க வேண்டும்.

ஊ) char test [4];

test = {“abc”};

ஓர் அணியில் இவ்வாறு மதிப்பிருத்த முடியாது. அறிவிக்கும் போதே தொடக்க மதிப்பிருத்தி,

char test [4] = “abc”;

எனக் குறிப்பிடலாம். அல்லது,

char test [4];

strcpy (test, “abc”);

எனச் சரமதிப்பை, சர மாறியில் படியெடுக்கலாம்.

எ) int num[ ] = {1,2,3}, num2 [3];

num2 = num;

அணிகளைக் குழுவாக அறிவித்து மதிப்பிருத்த வழியில்லை.

int num1[5], num2[10], num3[15];

என வெறுமனே குழுவாக அறிவிக்கலாம். தொடக்க மதிப்பிருத்தலும் செய்ய வேண்டியிருப்பின் தனித்தனியேதான் செய்ய வேண்டும்.

ஏ) int num [3];

cout << num;

cin >> num;

இது போன்ற அணி உள்ளீட்டு / வெளியீட்டுச் செயல்பாடுகளுக்கு அனுமதியில்லை. ஒவ்வோர் உறுப்பாகவே கையாள முடியும்.

cout << num [1];

cin >> num [2];

என அமைக்கலாம்.

ஐ) int roster = {1,2,3,4};

-roster என்னும் மாறி ஒரேயொரு மதிப்பையே ஏற்கும். எனவே,

int roster = 10; அல்லது, int roster [ ] = {1,2,3,4};

என அமைக்கலாம்.

2. கீழ்க்காணுமாறு தொடக்க மதிப்பிருத்தப்பட்டபின் அணியின் உள்ளடக்கம் என்னவாக இருக்கும்?

அ) int rate [ ] = {30,40,50};

ஆ) char ch [b] = {" bbbbb\0 " } -b என்பது இடவெளியைக் குறிக்கிறது.

ch [0] = 'C';

ch [4] = 'T';

ch [3] = 'A';

இ) char product-list [ ] [5] = {"nuts", "Bolts", "Screw"};

விடைகள் :

அ) rate [0] = 30, rate [1] = 40. rate [2] = 50

ஆ) ch [0] = 'C', ch [1] = ' '; ch [2] = ' ', ch [3] = 'A', ch [4] = 'T', ch [5] = '\ 0',

இ) product-list [0] = "Nuts \ 0", product-list [1] = "Bolts \ 0",

product-list [2] = "Screw\0".

3. கீழ்க்காணும் நிரல்களின் வெளியீடு என்னவாக இருக்கும்?

அ) விடை: END

```
# include <iostream.h>
void main ( )
{char ch [ ] = "END";
cout << ch;
}
```

ஆ)

```
# include <iostream.h>
void main ( )
```

```
{int a [ ] = {1,2,3,4,5};
for (int i = 0; i < 4; i++)
a[i+1] = a[i];
for (i= 0; i<5; i++)
cout << '\n' << a[i];
```

விடை

அணியின் தொடக்க உள்ளடக்கம் :

a[0]=1, a[1]=2,a[2]=3,a[3]=4,a[4]=5

முதல் for() மடக்கு செயல்படும் போது,

i=0:a[1]= a[0]

i=1:a[2]=a[1]

i=2:a[3]=a[2]

i=3:a[4]=a[3]

எனவே வெளியீடு,

1

1

1

1

1

என அமையும்

c)
```
# include <iostream.h>
# include <conio.h>
void main ( )
{ char name [ ] = "Jerry "; int k = 5;
        for (int i = 0 ; i < 3; i ++, k —)
                name [k] = name [i];
        cout << name;
        getch ( );
}
```

விடை :

for() மடக்கு செயல்படும்போது,

i = 0: k = 5,   name [5] = name [0];

i = 1: k = 4,   name [4] = name [1];

i = 2: k = 3,   name [3] = name [2];

எனவே, வெளியீடு, jerreJ எனக் கிடைக்கும்.

4. நிரல்கள் எழுதுக.

1) int-array என்னும் ஓர் int அணியை 10, 20, 30, 40, 50 ஆகிய தொடக்க மதிப்புகளுடன் வரையறுக்கவும். இந்த அணி உறுப்புகளின் மதிப்புகளைக் கூட்டி விடையைக் காட்டவும்.

2) 10 மதிப்புகளை இருத்திவைக்க கூடிய ஓர் int அணியை அறிவிக்கவும். அணியின் உறுப்புகளுக்குப் பயனரிடமிருந்து மதிப்புகளைப் பெறுக. அம்மதிப்புகளை முன்பின் வரிசையில் திரையிடுக.

3) word என்னும் சர மாறியில் (char அணி) பயனரிடமிருந்து ஒரு சொல்தொடரைப் பெற்று இருத்துக. while() மடக்கு switch() கூற்று ஆகியவை பயன்படுத்தித் தொடரிலுள்ள உயிரெழுத்துகளைக் (vowels) கணக்கிட்டு விடையை வெளியிடுக. எடுத்துக்காட்டாக,

```
char word [ ] = " The vowel count AEIOU aeiou";

vowel count is 14
```

4) MATRIX[3][3] என்னும் அணிக்கோவையை உருவாக்கவும். மூலைவிட்ட உறுப்புகளின் மதிப்புகளையும் அவற்றின் கூட்டுத் தொகையையும் வெளியிடுக.

5) matrixA[4][4], matrixB[4][4] ஆகிய இரு அணிக்கோவைகளுக்கு மதிப்புகளை உள்ளிடாகப் பெறுக. இரண்டு அணிக்கோவைகளையும் கூட்டி sum–matrix[4][4] என்னும் அணிக்கோவையில் இருத்தி, அதன் உள்ளடக்கத்தை வெளியிடுக.

எடுத்துக்காட்டு

matrix A				**matrix B**				**sum-matrix**			
1	2	3	4	9	8	7	6	10	10	10	10
5	6	7	8	5	4	3	2	10	10	10	10
9	1	2	3	1	9	8	7	10	10	10	10
4	5	6	7	6	5	4	3	10	10	10	10

அதிகாரம் 15

இனக்குழுக்களும் பொருள்களும்

(Classes and Objects)

15.1 இனக்குழுக்கள் – ஒரு முன்னுரை

சி++ மொழியின் மிக முக்கியமான பண்புக்கூறு 'இனக்குழு' என்பதாகும். சி++ மொழியை உருவாக்கிய ஜேன் ஸ்ட்ரௌஸ்ட்ரப் தம் மொழிக்கு முதலில் இனக்குழுவுடன் கூடிய சி' (C with Classes) என்று தான் பெயர் சூட்டினார் என்கிற தகவல், இனக்குழுவின் முக்கியத்துவத்தை எடுத்தியம்புகிறது. பயனர் வரையறுக்கும் தரவினங்களுள் 'இனக்குழு' என்பது, பயனர்கள் புதிய தரவினங்களை உருவாக்கவும் நடைமுறைப்படுத்தவும் ஒரு புதிய வழியைத் திறக்கிறது. வேறுபட்ட இனத்தரவுகளை ஒன்றாகச் சேர்த்துவைக்க இனக்குழுக்கள் ஒரு புதிய வழிமுறையை வழங்குகின்றன.

```
class student
{
        char name[30];
        int rollno, marks1, marks2, total_marks;
}
```

மேற்கண்ட இனக்குழுவில் அறிவிக்கப்பட்டுள்ள rollno, marks1, marks2, total_marks ஆகிய தரவு மாறிகள் student என்னும் இனத்தின் பண்புக்கூறுகளை வெளிப்படுத்துகின்றன. இந்தத் தரவுகளைக் கையாளக் கூடிய செயற்கூறுகளை வரையறுத்து, student இனக்குழுவை மேலும் விரிவுபடுத்த முடியும். இத்தகைய செயற்கூறுகளை **வழிமுறைகள்** (methods) என்றும் கூறுவர். காரணம், தரவுகளின் மீது நிகழ்த்தக் கூடிய பல்வேறு செயல்பாடுகளை (தரவு மதிப்புகளை உள்ளீடாக பெறுதல், தரவுகள் பங்குபெறும் கணக்கிடுகள் போன்றவை) அவை வரையறை செய்கின்றன.

சுருங்கக் கூறின்,

தரவுகளையும் அவற்றோடு தொடர்புடைய செயற்கூறுகளையும் ஒன்றாகச் சேர்த்துவைக்க இனக்குழு வகை செய்கிறது.

15.2 இனக்குழுவை வரையறுத்தல்

ஓர் இனக்குழுவின் வரையறுப்பு இருபகுதிகளைக் கொண்டது.

1. இனக்குழு அறிவிப்பு
2. இனக்குழுவின் செயற்கூறு வரையறைகள்

```
// Program - 15.1
# include <iostream.h>
# include <conio.h>
class student
{
          private :
            char name[30];
            int rollno, marks1, marks2 ,total_marks;
          protected:
            void accept()
            {
                cout<<"\n Enter data name, roll no, marks 1 and marks 2.. ";
                cin>>name>>rollno>>marks1>>marks2;
            }
 void compute()
           {
                 total_marks = marks1+ marks2;
           }
          void display()
           {
                     cout<<"\n Name "<<name;
                    cout<<"\n Roll no "<<rollno;
                    cout<<"\n Marks 1.. "<<marks1;
                    cout<<"\n Marks 2.. "<<marks2;
                    cout<<"\n Total Marks.. "<< total_marks;
          }
        public:
        student()
        {
              name[0]='\0';
              rollno=marks1=marks2=total_marks= 0;
              cout<<"\n Constructor executed ... ";
        }
        void execute()
       {
              accept();
              compute();
              display();
      }
};
void main()
{
       clrscr();
       student stud;
       stud.execute();
}
```

இனக்குழு அறிவிப்பின் வடிவம்

பொதுவடிவம்	**மேற்கண்ட எடுத்துக்காட்டில் உள்ள இனக்குழு வடிவம்**

```
class class-name
{
  private:
    variable decl.
    function decl.

  protected:
    variable decl.
    function decl.

  public:
    variable declaration
    function declaration
};
```

```
class student
{ private:
    char name [10];
    int roll no, mark1, mark2, total marks;

protected:
    void accept( );
    void compute( );
    void display( );

public:
student( );
void execute( );
};
```

- ✓ class என்னும் சிறப்புச் சொல், இது பயனர் வரையறுக்கும்.class என்னும் தரவு இனம் என்பதை உணர்த்துகிறது
- ✓ இனக்குழுவின் உடற்பகுதி நெளிவு அடைப்புக் குறிகளால் அடைக்கப்பட்டு ஓர் அரைப்புள்ளியுடன் முற்றுப் பெறுகிறது.
- ✓ இனக்குழுவின் உடற்பகுதி மாறிகள் மற்றும் செயற்கூறுகளின் அறிவிப்புகளைக் கொண்டுள்ளது.
- ✓ இனக்குழுவின் உடற்பகுதி மூன்று அணுகியல்பு வரையறுப்பிகளைக் (வெளிக்காட்டும் சிட்டைகள்) கொண்டுள்ளது. private, public, protected.
- ✓ private என்னும் அணுகியல்பைக் குறிப்பிடுவது கட்டாயமில்லை. எவ்விதஅணுகியல்பும் குறிப்பிடப்படவில்லை எனில் இனக்குழுவின் உறுப்புகள் இயல்பாக private என்றே கருதிக் கொள்ளப்படும்.
- ✓ private என அறிவிக்கப்பட்ட உறுப்புகளைஅந்த இனக்குழுவுக்குள்ளே தான் அணுக முடியும்.
- ✓ Protected என அறிவிக்கப்பட்ட உறுப்புகளை அந்த இனக்குழுவுக்குள்ளும், அந்த இனக்குழுவின் அடிப்படையில் தருவிக்கப்பட்ட இனக்குழுவுக்குள்ளும் எடுத்தாள முடியும்.
- ✓ public என அறிவிக்கப்பட்ட உறுப்புகளை அந்த இனக்குழுவுக்கு வெளியேயும் எடுத்தாள முடியும்.

15.3 தரவு அருவமாக்கம் (Data Abstraction)

தரவுகளையும் செயற்கூறுகளையும் ஒன்று சேர்த்து ஒற்றை உருபொருளாய்க் கட்டிவைப்பதை **உறைபொதியாக்கம்** (Encapsulation) என்று அழைக்கிறோம்.

private என வகைப்படுத்தப்பட்ட தரவுகளையும் செயல் கூறுகளையும் இனக்குழுவுக்கு வெளியிலிருந்து அணுக முடியாது. இதனைத் **தரவு மறைப்பு**(Data hiding) என்கிறோம். பிற

இனக்குழுக்களின் பொருள்களும்(objects), உறுப்புகளும்(members) வரம்புக்குட்பட்ட முறையில் அணுகுமாறு கட்டுப்பாடு விதிக்கும் நுட்பம் **தரவு அருவமாக்கம்** (Data Abstraction) எனப்படுகிறது. தரவு மறைப்பு மூலமாக 'தரவு அருவமாக்கம்' சாத்தியமாகி உள்ளது என்றும் கூறலாம்.

தரவு மறைப்பு என்பது பொருள்நோக்கு நிரலாக்கத்தின் (Object Oriented Programming-OOP) அடிப்படையான பண்புக் கூறாகும்.

private	அதே இனக்குழுவின் உறுப்புகளும் நட்புச் செயற்கூறுகள் (friend functions)எனப்படும் சில சிறப்புச் செயற்கூறுகளும் மட்டுமே அணுக முடியும்
protected	அதே இனக்குழு மற்றும் அதன் தருவிக்கப்பட்ட இனக்குழுக்களின் உறுப்புகளால் அணுக முடியும்
public	அதே இனக்குழுவின் உறுப்புகளும் பொருள்களும் மட்டுமின்றி வெளி இனக்குழுக்களின் உறுப்புகளாலும் அணுக முடியும்

15.4 தரவு உறுப்புகளும் உறுப்புச் செயற்கூறுகளும்

இனக்குழு இருவகை உறுப்புகளைக் கொண்டது. ஒன்று **தரவு உறுப்புகள்**(Data members). மற்றது **உறுப்புச் செயற்கூறுகள்**(member functions). தரவு உறுப்புகள் என்பவை ஓர் இனக்குழுவின் பண்புக்கூறுகளை உணர்த்தும் தரவு மாறிகளைக் குறிக்கின்றன. உறுப்புச் செயற்கூறுகள் என்பவை ஓர் இனக்குழுவில் குறித்த பணிகளை நிறைவேற்றுகின்ற செயற்கூறுகளைக் குறிக்கின்றன. உறுப்புச் செயற்கூறுகளை வழிமுறைகள் (methods) என்றும் தரவு உறுப்புகளைப் பண்புக்கூறுகள் (attributes)என்றும் கூறுவது உண்டு. இப்போது, program-15.1- ல் வரையறுக்கப்பட்டுள்ள student என்னும் இனக்குழுவை அடிப்படையாகக் கொண்டு தகவல் தரப்பட்டுள்ள அட்டவணை 15.1ஐ நோக்குக. இனக்குழுக்கள் **ஆக்கிகள்** (constructors), **அழிப்பிகள்**(destructors) என்னும் தனிச் சிறப்பான உறுப்புச் செயற்கூறுகளையும் கொண்டுள்ளன. ஆக்கிகள் /அழிப்பிகள் பற்றிப் பாடம் 17ல் விரிவாகப் படிப்போம்.

student	தரவினக் குறிப்பெயர் student என்பதை இனக்குழு ஒட்டு (tag) என்றும் கூறுவர்
name, rollno, marks1, marks2, total_ marks	தரவு உறுப்புகள்
accept ( ) compute ( ) display ( ) execute ( ) student ( )	உறுப்புச் செயற்கூறுகள் அல்லது வழிமுறைகள்
stud	student இனக்குழுவின் சான்றுரு (instance)/பொருள் (object)/ மாறி (variable)

அட்டவணை15.1 Program - 15.1 **ல் உள்ள** student **இனக்குழு**

15.5 பொருள்களை உருவாக்குதல்

student stud;

என்னும் அறிவிப்புக் கூற்றை நோக்குங்கள். இக்கூற்றின் பொருள் stud என்பது student இனக்குழுவின் சான்றுரு(instance) அல்லது பொருள் (object) என்பதாகும்.

ஓர் இனக்குழு வரையறுக்கப்பட்டபின், அவ்வினத்தில் மாறிகளை அறிவிக்க முடியும். stud என்பது student இனத்தைச் சார்ந்த மாறி, student என்பது class என்னும் தரவினமாகும். சி++ மொழியில் இனக்குழு மாறிகள் **பொருள்கள்** (object) என்று அறியப்படுகின்றன. அடிப்படைத் தரவினங்களில் மாறிகளை அறிவிப்பதைப் போன்றே இனக்குழுவில் பொருள்கள் அறிவிக்கப்படுகின்றன. இனக்குழு வரையறுப்பு முடிகின்ற நெளிவு அடைப்புக் குறியை ஒட்டி, மாறிகளின் பெயரைக் குறிப்பிட்டும் பொருள்களை உருவாக்க முடியும்.

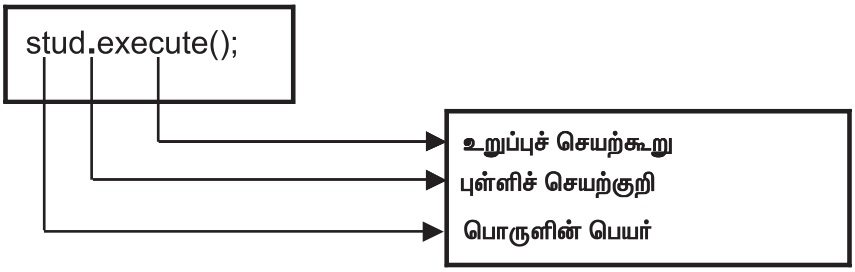
```
class student
{
        private:


          protected:


          public:

}stud;// stud is an object
```

```
class student
{
        private:
        ……..
        protected:
        …….
         public:
};
void main()
{
   student s, s1[5];
}
the variables s and s1 are objects
or instances of the class stduent
```

படம் 15.1 பொருள்களை உருவாக்குதல்

15.6 இனக்குழு உறுப்புகளை அணுகுதல்

இனக்குழுவின் உறுப்புகளைப் புள்ளிச் செயற்குறி மூலம் அணுக முடியும். எடுத்துக்காட்டாக, student இனக்குழுவின் excute() செயற்கூறினை அழைக்கும் கூற்று இவ்வாறு அமையும்.



ஓர் இனக்குழுவின் private தரவுகளை அதே இனக்குழுவின் உறுப்புச் செயற்கூறுகளும், நட்புச் செயல்கூறுகள் எனப்படும் சில சிறப்புச் செயற்கூறுகளுமே அணுக முடியும்.

எடுத்துக்காட்டில் தரப்பட்டுள்ள student இனக்குழுவில் name,marks1, marks2, total_marks ஆகிய தரவு உறுப்புகளை accept(), display(), compute() ஆகிய உறுப்புச் செயற்கூறுகள் மட்டுமே அணுக முடியும். இனக்குழுவுக்கு வெளியே அறிவிக்கப்படும் பொருள்கள் private அல்லது Protected என வரையறுக்கப்பட்டள்ள உறுப்புகளையோ, செயற்கூறுகளையோ அணுக முடியாது.

Public என அறிவிக்கப்பட்டுள்ள உறுப்புச் செயற்கூறுகளை அதே இனக்குழுவின் பொருள்களால் அணுக முடியும்.

stud. execute();

என்னும் அழைப்புக் கூற்று ஏற்கத் தக்கதே. காரணம், execute() என்பது public என அறிவிக்கப்பட்டுள்ள ஓர் உறுப்புச் செயற்கூறு. எனவே, இனக்குழுவுக்கு வெளியே அறிவிக்கப்பட்டுள்ள stud என்னும் பொருள் மூலமாக அணுக முடியும். ஆனால் stud. accept(), stud.compute(). stud.display(), stud.marks1 - என்றெல்லாம் அணுக முயன்றால், நிரல்பெயர்ப்பி "not accessible" என்னும் பிழைசுட்டும்செய்தியைத் தரும். இரண்டு எண்களைக் கூட்டி சொல்லும் பணியை விளக்கும் Program-15.2 இன்னோர் எடுத்துக் காட்டாகும். இதிலுள்ள இனக்குழு, மூன்று முழுஎண் மாறிகளையும், எண்களைப் பெற்றுக் கூட்டித் தருகின்ற செயற்கூறினையும் கொண்டுள்ளது sum என்னும் மாறி public என்னும் அணுகியல்பு கொண்டுள்ளதால், s என்னும் பொருள் அதனை அணுக முடிகிறது.

```
//Program - 15.2
# include <iostream.h>
# include <conio.h>
class add
{
                private:
                        int a,b;
                public:
                        int sum;

                        void getdata()
                        {
                              a=5;
                              b=10;
                              sum = a+b;
};
void main()
{
   add s;
   s.getdata();
   cout<<s.sum;
}
```

15.7 இனக்குழுவின் செயற்கூறுகளை வரையறுத்தல்

```
class add
{
        int a,b;
        public:
         add()
         {
                 a= 0;
                 b= 0;
         }
  void display();

  };
void add::display()
  {
    int sum;
    sum = a+b;
    cout<<sum;

  }
```

வழிமுறை 1 (add() { a= 0; b= 0; })

வழிமுறை–2 (void add::display() { int sum; sum = a+b; cout<<sum; })

வழிமுறை 1–ல், add() என்னும் உறுப்புச் செயற்கூறு, add() இனக்குழுவுக்குள்ளேயே அறிவிக்கப்பட்டு, வரையறுக்கப்பட்டுள்ளது.

வழிமுறை 2–ல், display() என்னும் உறுப்புச் செயற்கூறு, add இனக்குழுவுக்குள் அறிவிக்கப்பட்டு, இனக்குழுவுக்கு வெளியே வரையறுக்கப்பட்டுள்ளது.

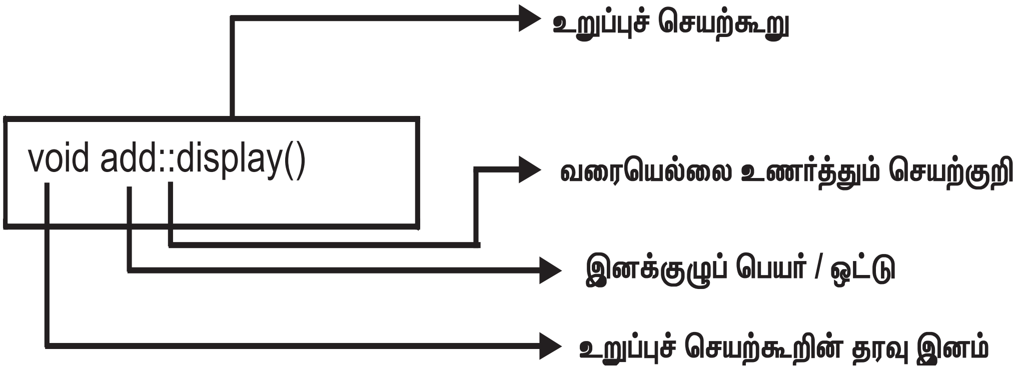
இனக்குழுவின் உறுப்புச் செயற்கூறுகளை இவ்வாறு இரு வகையாகவும் வரையறுக்கலாம். இனக்குழுவுக்குள் வரையறுக்கப்படும் செயற்கூறுகள் inline செயற்கூறுகளைப்போல் இயங்குகின்றன.

இனக்குழுவுக்கு வெளியே வரையறுக்கப்படும் செயற்கூறுகள்,

type class_name :: function name();

என்னும் முன்வடிவைக் கொண்டுள்ளன.

எடுத்துக்காட்டு



function_name()-க்கு முன்பாக இடம் பெற்றுள்ள class_name::(add::) என்னும் உறுப்பைக் குறிக்கும் முன்னொட்டு, அச்செயற்கூறு class_name -க்கு உரியது என்பதைச் சுட்டுகிறது. செயற்கூறின் தலைப்பில் குறிப்பிடப்பட்டுள்ள இனக்குழுவுக்குள் அடங்கியது என்னும் வரையெல்லையைக் குறிக்கிறது.

உறுப்புச் செயற்கூறுகள் தனிச் சிறப்பான பண்பியல்புகளைக் கொண்டுள்ளன. நிரல் உருவாக்கத்தில் இவை அடிக்கடி பயன்படுத்திக் கொள்ளப்படுகின்றன.

- ✓ பல இனக்குழுக்கள் தம் செயற்கூறுகளுக்கு ஒரே பெயரைப் பயன்படுத்திக் கொள்ள முடியும். செயற்கூறினுக்கு முன்னொட்டாக இடம்பெறும் இனக்குழுப் பெயர் அதன் வரையெல்லையைக் குறிக்கும்.
- ✓ உறுப்புச் செயற்கூறுகள் இனக்குழுவின் private தரவுகளை அணுக முடியும். உறுப்பில்லாத செயற்கூறுகள் அணுக முடியாது.
- ✓ ஓர் உறுப்புச் செயற்கூறு இன்னோர் உறுப்புச் செயற்கூறினை நேரடியாக அழைக்க முடியும். புள்ளிச் செயற்குறி தேவையில்லை. (இவ்வாறு அமைவது உறுப்புச் செயற்கூறுகளின் பின்னலமைப்பு எனப்படுகிறது.)
- ✓ உறுப்புச் செயற்கூறுகள், ஏற்கத்தகு சி++ தரவு இனச் செயலுருபுகளை ஏற்கும். பொருள்களையும் செயலுருபுகளாக அனுப்பிவைக்க முடியும்.
- ✓ உறுப்புச் செயற்கூறு திருப்பியனுப்பும் மதிப்பு, இனக்குழுவின் பொருளாகவும் இருக்கலாம்
- ✓ உறுப்புச் செயற்கூறுகள் static இனமாகவும் இருக்க முடியும்.

15.8 பொருள்களுக்கு நினைவக ஒதுக்கீடு

இனக்குழு வரையறுப்பின் ஒரு பகுதியாக உறுப்புச் செயற்கூறுகள் வரையறுக்கப்பட்டிருக்கும் போதே அவை உருவாக்கப்பட்டு நினைவகத்தில் இருத்தப்படுகின்றன. அந்த இனக்குழுவைச் சார்ந்த அனைத்துப் பொருள்களுமே ஒரே உறுப்புச் செயற்கூறினையே பயன்படுத்திக் கொள்வதால், ஒவ்வொரு பொருள் உருவாக்கப்படும் போதும் உறுப்புச் செயற்கூறுகளுக்குத் தனித்தனி நினைவக இடம் ஒதுக்கப்படுவதில்லை. உறுப்பு மாறிகளுக்குத் தேவையான நினைவக இடம் மட்டும் ஒவ்வொரு பொருளுக்கும் தனித்தனியே ஒதுக்கப்படுகிறது. வெவ்றேு பொருள்களுக்கு வெவ்வேறு இடம் ஒதுக்கப்படுவது கட்டாயமாகும். காரணம், வெவ்வேறு பொருள்களின் உறுப்புமாறிகள் வெவ்வேறு மதிப்புகளைக் கொண்டிருக்கும்.

கீழேயுள்ள இனக்குழு அறிவிப்பை நோக்குக.

```
class product
{
        int code, quantity;
        float price;
        public:
                void assign_data();
                void display();
```

```
};
void main()
{
product p1, p2;
}
```

இதிலுள்ள உறுப்புச் செயற்கூறுகள் assign_data(), display() ஆகியவை நினைவகத்தில் பொதுவிடப் பகுதியில் இருத்தப்படுகின்றன. அதாவது p1, p2 ஆகிய இரண்டு பொருள்களும் பொதுவிடப் பகுதியிலுள்ள கட்டளைகளை அணுக முடியும்.

p1, p2 ஆகிய இரண்டு பொருள்களின் நினைவக ஒதுக்கீட்டைக் காண்க.

பொருள்	தரவு உறுப்புகள்	நினைவக ஒதுக்கீடு
p1	Code, quantity, price	8 பைட்டுகள்
p2	Code,quantity, price	8 பைட்டுகள்

அட்டவணை 15.2 பொருள்களுக்கான நினைவக ஒதுக்கீடு

ஓர் இனக்குழுவின் உறுப்புச் செயற்கூறுகள், program 15.3ல் விளக்கப்பட்டுள்ளதுபோல, உறுப்பில்லாத வேறெந்தச் செயற்கூறுகளையும் போலவே செயலுருபுகளைக் கையாள முடியும்.

15.9 Static தரவு உறுப்புகள்

இனக்குழுவின் ஒரு தரவு உறுப்புக்கு static என்னும் தகுதியை அளிக்க முடியும். static தரவு உறுப்பு.

- ✓ அந்த இனக்குழுவில் முதல் பொருள் உருவாக்கப்படும்போது, 0 என்ற தொடக்க மதிப்பைப் பெறும், அதன் பிறகு தொடக்க மதிப்பிருத்தப்படுவதில்லை.
- ✓ நினைவகத்தில், உறுப்பு மாறிக்கு ஒரேயொரு நகல் மட்டுமே இருக்கும் (பொது இடத்தின் ஒரு பகுதியாக) அதே இனக்குழுவின் பிற பொருள்களும் நினைவகத்திலுள்ள அந்தவொரு நகலையே பகிர்ந்து கொள்கின்றன.
- ✓ அதன் வரையெல்லையும் அணுகியல்பும் அந்த இனக்குழுவுக்குள்ளே தான். ஆனால் அதன் வாழ்நாள், நிரல் செயல்பட்டு முடியும்வரை இருக்கும்.

static உறுப்பு மாறி பயன்படுத்தப்பட்டுள்ள ஒரு நிரலைக் காண்க. (program-15.4):

```
// Program - 15.3
#include<iostream.h>
#include<conio.h>

class product
{
        int code, quantity;
        float price;
public:
        void assign_data( int c, int q, float p)
        {
         code = c;
         quantity = q;
         price = p;
         }
        void display()
        {
         cout<<"\n Code : "<<code;
         cout<<"\n Quantity :"<<quantity;
         cout<<"\n Price : "<< price;
         }
};
void main()
{
          product p;
          p.assign_data( 101, 200, 12.5);
          p.display();
}
```

```
// Program - 15.4
// To demonstrate the use of static member variables
#include<iostream.h>
#include<conio.h>
class simple_static
{
                int a,b,sum;
                static int count;
public:
                 void accept()
                 {
                        cout<<“\n Enter values.. “;
                        cin>>a>>b;
                        sum = a+b;
                        count++;
                }
                void display()
               {
                        cout<<“\n The sum of two numbers … “<<sum;
                        cout<<“\n This is addition… “<<count;
}
};
int static_ simple.count=0;
void main()
{
             simple_static p1,p2,p3;
             p1.accept();
             p1.display();
             p2.accept();
             p2.display();
             p3.accept();
             p3.display();
}
```

வெளியீடு:

Enter values …… 10 20

The sum of two numbers ………… 30

This is addition 1

Enter values……… 5 7

The sum of two numbers……………12

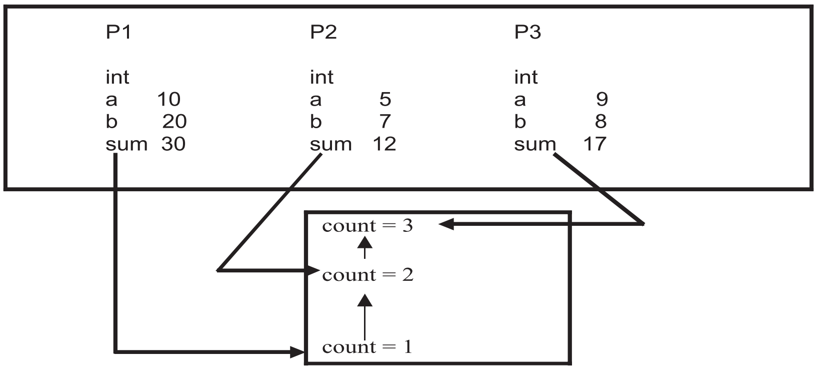
This is addition 2

Enter values……….. 9 8

The sum of two numbers ……………17

This is addition 3

static மாறியான count ஒரேயொரு முறை மட்டும் 0 எனத் தொடக்க மதிப்பிருத்தப்படுகிறது. ஒவ்வொரு முறை இரண்டு எண்களின் கூட்டுத்தொகை கணக்கிடப்படும் போதெல்லாம் count ன் மதிப்பு ஒன்று மிகுக்கப்படுகிறது. accept() செயற்கூறு மும்முறை செயல்படுத்தப்படுவதால் count-ன் மதிப்பும் மும்முறை மிகுக்கப்பட்டு 3 என்னும் மதிப்பைப் பெறுகிறது. அனைத்துப் பொருள்களும் count - ன் ஒரேயொரு நகலை மட்டுமே பகிர்ந்து கொள்வதால், count - ன் மதிப்பு 3 ஆகிறது. இது, படம் 15.2ல் விளக்கப்பட்டுள்ளது.



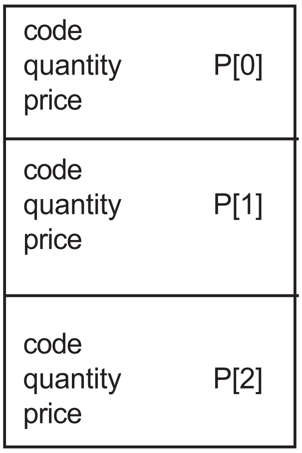
படம்15.2 Static **உறுப்பு மாறி** - Count

static உறுப்பு மாறியின் தொடக்க மதிப்பு இனக்குழுவுக்கு வெளியே இருத்தப்படுகிறது.

15.10 பொருள்களின் அணிகள்

கீழ்க்காணும் இனக்குழு வரையறுப்பையும், அதன் நினைவக ஒதுக்கீட்டையும் நோக்குக.

```
class product
{
   int code,quantity;
   float price;
public :
   void assign_Data();
   void display();
} p[3];

void main()
{
  p[0].assign_Data();
  p[0].display();
}
```



பயிற்சி வினாக்கள்

I. பிழைகளைக் கண்டறிந்து திருத்துக

```
class x
{
        public:
                int a,b;
                void init()
                {
                        a =b = 0;
                }
                int sum();
                int square();
};
int sum()
        {
                return a+b;
        }
        int square()
        {
                        return sum() * sum()
        }
```

விடை :

int x::sum() மற்றும் int x::square()

II

```
#include<iostream.h>
class simple
{
                int num1, num2 , sum = 0;
        protected:
                accept()
                {
                        cin>>num1>>num2;
                }
        public:
        display()
        {
                sum = num1 + num2;
        }
};
```

```
void main()
{       simple s;
        s.num1=s.num2= 0;
        s.accept();
        display();
}
```

விடை:

1. sum என்னும் உறுப்பு மாறியில் அறிவிப்பின்போது தொடக்க மதிப்பு இருத்த முடியாது.
2. num1,num2 ஆகிய உறுப்பு மாறிகள் private என்பதால் அவற்றை main() லிருந்து அணுக முடியாது.
3. s.accept() என்று அழைக்க முடியாது. காரணம் அது protected என வரையறுக்கப்பட்டுள்ளது.
4. display() என்னும் செயற்கூறினை ஒரு பொருளின் மூலம் அழைக்க வேண்டும்.

III

```
#include<iostream.h>
#include<conio.h>
class item
{
        private:
        int code,quantity;
        float price;
        void getdata()
        {
                cout<<"\n Enter code, quantity, price ";
                cin>>code>>quantity>>price;
        }
        public:
                float tax=0;
                void putdata()
                {
                        cout<<"\n Code : "<<code;
                        cout<<"\n Quantity : "<<quantity;
```

```
                    cout<<"\n Price : "<<price;
                    if( quantity >100 )
                            tax = 2500;
                    else
                            tax =1000;
                            cout<<" \n Tax :"<<tax;
                    }
};
void main()
{ item i;
}
```

மேற்கண்ட நிரலின் அடிப்படையில் கீழ்க்காணும் அட்டவணையை நிறைவு செய்க.

சான்றுரு i–ன் நினைவக ஒதுக்கீடு	private தரவு உறுப்புகள்	public தரவு உறுப்புகள்	i மூலம் அணுக முடிகிற செயற் கூறுகளும் தரவு உறுப்புகளும்

IV

1) கீழ்க்காணும் வரையறுப்புகளுடன் employee என்னும் ஓர் இனக்குழுவை வரையறுக்கவும்:

employee இனக்குழுவின் private உறுப்புகள் :

empno- முழுஎண்

ename – 20 எழுத்துகள்

basic - float

netpay, hra, da, float

calculate () – basic+hra+da என்பதைக் கணக்கிட்டு ஒரு float இனமதிப்பைத் திருப்பியனுப்பும் செயற்கூறு.

employee இனக்குழுவின் public உறுப்புச் செயற்கூறுகள் :

havedata() – empno, ename, basic, hra, da –ஆகிய மதிப்புகளை உள்ளீடாகப் பெற்று, calculate() செயற்கூறினை அழைத்து

netpay – ஐக் கணக்கிடச் செய்யும் செயற்கூறு.

dispdata() – தரவு உறுப்புகள் அனைத்தையும் திரையில் காட்டும் செயற்கூறு.

2) கீழ்க்காணும் வரையறுப்புகளுடன் math என்னும் இனக்குழவை வரையறுக்கவும்

private உறுப்புகள் :

num1, num2, result – float

init() செயற்கூறு– num1, num2, result ஆகியவற்றில் 0 என்னும் தொடக்க மதிப்பிருத்தும்

protected உறுப்புகள் :

add() செயற்கூறு – num1, num2 இரண்டையும் கூட்டி, கூட்டுத் தொகையை result–ல் இருத்தும்.

prod() செயற்கூறு –num1, num2 இரண்டையும் பெருக்கி, பெருக்குத் தொகையை result–ல் இருத்தும்.

public உறுப்புகள்:

getdata() செயற்கூறு – num1 , num2 ஆகிய மதிப்புகளை ஏற்கும்.

menu() செயற்கூறு– 1. Add…

2. Prod…

என்னும் பட்டியைக் (menu) காட்டும்.

1 என்பதைத் தேர்வு செய்தால் add() செயற்கூறினையும், 2ஐத் தேர்வு செய்தால் prod() செயற்கூறினையும் செயல்படுத்தி, result - ஐத் திரையில் காட்ட வேண்டும்.

அதிகாரம் 16

பல்லுருவாக்கம்

16.1 முன்னுரை

polymorphism என்ற சொல், ‘பல வடிவங்கள்’ (poly-many, morph-shapes) என்னும் பொருளைத் தருகிறது. ‘தமிழில் **பல்லுருவாக்கம்** என்கிறோம். சி++ மொழியில், பல்லுருவாக்கம் **செயற்கூறு பணிமிகுப்பு** (function overloading) **செயற்குறி பணிமிகுப்பு** ஆகியவற்றின் மூலம் நிறைவேற்றப்படுகிறது. உண்மையில் ‘பணிமிகுப்பு’ (overloading) என்பது ஒரே பெயர். ஒன்றுக்கு மேற்பட்ட வெவ்வேறு பொருளை உணர்த்துவதைக் குறிக்கிறது. ஆக, பணிமிகுத்த செயற்கூறு என்பது ஒன்றுக்கு மேற்பட்ட தனித்த பொருள் கொண்ட செயற்கூறினைக் குறிக்கிறது. பொருள்நோக்கு நிரலாக்கத்துக்குத் துணை புரியும் ‘செயற்கூறு பணிமிகுப்பு’ சி++ மொழியின் பண்புக்கூறுகளில் ஒன்றாகும்.

16.2 செயற்கூறு பணிமிகுப்பு (Function overloading)

செய்தி அல்லது தரவினை ஒன்றுக்கு மேற்பட்ட வடிவங்களில் செயலாக்கவல்ல செயற்கூறின் திறனையே செயற்கூறு பணி மிகுப்பு என்கிறோம்.

ஒரு நிரலர் கீழ்க்காணும் செயற்கூறுகளை வரையறுக்க விரும்புகிறார் என வைத்துக் கொள்வோம்.

```
area_circle()       // ஒரு வட்டத்தின் பரப்பைக் கணிக்க
area_triangle()     // ஒரு முக்கோணத்தின் பரப்பைக் கணிக்க
area_rectangle()    // ஒரு செவ்வகத்தின் பரப்பைக் கணிக்க
```

வெவ்வேறு வடிவங்களின் பரப்பளவைக் கண்டறிய எழுதப்பட்ட மேற்கண்ட மூன்று முன்வடிவுகளையும், ஒரே பெயரில் (வெவ்வேறு அளபுருக்களுடன்) வரையறுக்க முடியும்.

```
float area ( float radius);
float area ( float half, float base, float height );
float area ( float length , float breadth);
```

இப்போது program-16.1-ஐ நோக்குக, மூன்று வடிவங்களில் எந்த வடிவத்தின் பரப்பளவையும் கணக்கிட area() என்னும் செயற்கூறினைச் செயல்படுத்தியுள்ளமை காண்க.

```
// Program - 16.1
// to demonstrate the polymorphism - function overloading
#include<iostream.h>
#include<conio.h>
        float area ( float radius )
{       cout << "\nCircle …";
        return ( 22/7 * radius * radius );
}
float area (float half, float base, float height)
{       cout << "\nTriangle ..";
        return (half* base*height);
}
float area ( float length, float breadth )
{       cout << "\nRectangle …";
        return ( length *breadth ) ;
}
void main()
{
        clrscr();
        float r,b,h;
        int choice = 0 ;
        do
        {
                clrscr();
                cout << "\n Area Menu ";
                cout << "\n 1. Circle ... ";
                cout << "\n 2. Traingle ...";
                cout << "\n 3. Rectangle ... ";
                cout <<"\n 4. Exit ... ";
                cin>> choice;
                switch(choice)
                {
                        case 1 :
                                cout << "\n Enter radius ... ";
                                cin>>r;
                                cout<<"\n The area of circle is ... " << area(r);
                                getch();
                                break;
                        case 2:
                                cout<< "\n Enter base, height ... ";
                                cin>>b>>h;
                                cout<<"\n The area of a triangle is .. " << area (0.5, b, h);
                                getch();
                                break;
                        case 3:
                                cout<< "\n Enter length, breadth.. ";
                                cin>>h>>b;
                                cout<<"\n The area of a rectangle is ... " << area(h,b);
                                        getch();
                                        break;
        }
        }while (choice <=3);
}
```

மூன்று செயற்கூறுகளின் முன்வடிவுகளைக் கவனித்தீர்களா ? அவை

```
float area ( float radius );
float area (float half, float base, float height);
float area ( float length, float breadth );
```

மூன்று area() செயற்கூறுகளின் வரையறைகளும் ஒன்றுக்கொன்று மாறுபட்டுள்ளன என்று நினைக்கிறீர்களா ? ஆம், ஒவ்வொரு செயற்கூறின் முன் வடிவும் செயலுருபுகளின் எண்ணிக்கையில் வேறுபடுகின்றன. முதலாவது முன்வடிவில் ஒரு செயலுருபும், இரண்டாவதில் மூன்று செயலுருபுகளும், மூன்றாவதில் இரண்டு செயலுருபுகளும் உள்ளன. நம்முடைய எடுத்துக்காட்டில் மூன்று செயற்கூறுகளுமே float இனச் செயலுருபுகளைக் கொண்டுள்ளன. இவ்வாறுதான் இருக்க வேண்டும் என்பதில்லை. ஒவ்வொரு முன் வடிவும் வெவ்வேறு தரவினச் செயலுருபுகளைக் கொண்டிருக்கலாம். செயலருபுகளின் எண்ணிக்கை வெவ்வேறாக இருக்கலாம். செயலருபுகளின் எண்ணிக்கை ஒன்றாக இருப்பின் அவற்றின் தரவினங்கள் மாறி இருக்க வேண்டும்.

கீழ்க்காணும் முன்வடிவுகளில் செயற்கூறு பணிமிகுப்புக்கான முன்வடிவுகளில் பிழையானதைக் கண்டறிய முடியுமா ? ஏன் என்று காரணம் கூற முடியுமா ?

செயற்கூறு முன் வடிவுகள்:	பிழையான முன்வடிவுகள்:

void fun(int x);
void fun(char ch);
void fun(int y);
void fun(double d);
	void fun(int x);
	void fun(int y);
	இரண்டும் ஒரே எண்ணிக்கையில் ஒரே இனச் செயலுருபுகளைக் கொண்டுள்ளன.
	எனவே பிழையாகும்.

செயற்கூறு பணிமிகுப்பில் செயற்கூறுகள் எவ்வாறு இயக்கப்படுகின்றன ?

நிரல்பெயர்ப்பி, **மிகச் சிறந்த பொருத்தம்** (Best Match)என்னும் செயல்நுட்பத்தைப் பின்பற்றுகிறது. இந்த செயல்நுட்பத்தின்படி நிரல் பெயர்ப்பி.

- ✓ செயற்கூறு அழைப்புக் கூற்றுக்கு மிகச் சரியாகப் பொருந்தும் செயற்கூறு முன்வடிவைத் தேடி பார்க்கும்.
- ✓ மிகச் சரியாகப் பொருந்தும் செயற்கூறு இல்லையெனில், அடுத்துப் பொருத்தமுள்ள செயற்கூறினைத் தேடும். அதாவது, எண்வகைச் செயலுருபை வேறு எண்வகைக்கு இனமாற்றம் செய்து, அதற்கேற்ற செயற்கூறு முன்வடிவு உள்ளதா எனப் பார்க்கும்.

எடுத்துக்காட்டாக, மேற்கண்ட நிரலில் (program-16.1)float area (float radius) என்னும் முன்வடிவு உள்ளது, இதன் பொருள், அழைப்புக் கூற்று area(r) என்பதில் r என்னும் செயலருபு float இனத்தைச் சேர்ந்ததாக இருக்க வேண்டும். r என்பது int இனமாக இருக்கும் எனில், முழுஎண் வகை இனமாற்றத்தின்படி, int மாறிலி/மாறியை char அல்லது float அல்லது double இனத்தோடு பொருத்திக் கொள்ள முடியும். இச்செயல்நுட்பத்தின்படி area(r) என்னும் அழைப்பு area(float radius)என்னும் செயற்கூறோடு பொருத்தப்பட்டு அது இயக்கப்படும்.

இத்தகைய எண்வகை இனமாற்றம் முழுக்க முழுக்க நிரல்பெயர்ப்பி சார்பானது. வெவ்வேறு நிரல்பெயர்ப்பிகள் வெவ்வேறு வகையில் இனமாற்றம் செய்யலாம். எண்வகை இனமாற்றம் பெரும்பாலும் இவ்வாறு அமையும்.

✓ char தரவினத்தை int/float/double இனமாக மாற்ற முடியும்

✓ int தரவினத்தை char/float/double இனமாக மாற்ற முடியும்

✓ float தரவினத்தை int/double/char இனமாக மாற்ற முடியும்.

✓ double தரவினத்தை int அல்லது float இனமாக மாற்ற முடியும்.

program-16.1 நிரலில் உள்ள area() செயற்கூறினுக்கு அழைப்புக் கூற்று கீழ்க்காணுமாறு அமையுமெனில் விடை என்னவாக இருக்கும் ?

செயற்கூறு அழைப்புக் கூற்று	வெளியீடு
area(5.0)	
area(0.5,4.0,6.0)	
area(3.0,4.5)	

செயற்கூறு பணிமிகுப்பின் விதிமுறைகள்

1. பணிமிகுத்த செயற்கூறுகள் முறையான அளபுருக்களின் எண்ணிக்கையிலோ, அவற்றின் தரவு இனங்களிலோ வேறுபட்டிருக்க வேண்டும்.

2. பணிமிகுத்த செயற்கூறுகள் திருப்பியனுப்பும் தரவினம் ஒன்றாக இருக்கலாம், வேறுபட்டும் இருக்கலாம்.

3. பணிமிகுத்த செயற்கூறுகளின் முன்னியல்புச் செயலுருபுகளை, அளபுருக்களின் பட்டியலில் ஒரு பகுதியாக சி++ நிரல்பெயர்ப்பி கருதிக் கொள்ளாது.

4. தொடர்பில்லாத இரு செயற்கூறுகளுக்கு ஒரே பெயரைச் சூட்டாமல் இருப்பது நல்லது.

முறையற்ற அறிவிப்புகள், கீழேயுள்ளவாறு செயற்கூறு அழைப்பினில் மோதலை உருவாக்கும்.

```
void fun ( char a, int times)
{
       for (int i=1; i<=times;i++)
                     cout<<a;
}
void fun( char a= '*', int times )
{
       for(int i=1;i<=times;i++)
                   cout<<a;
}
void fun( int times)
{
       for(int i=1; i<=times ;i++)
                    cout<<'@';
}
void main()
{
       fun ( '+', 60);
       fun(60);
}
```

மேற்கண்ட நிரலை நிரல்பெயர்க்கும் போது இரண்டு பிழைகள் சுட்டப்படும்.

- ✓ Conflict between fun(char a, int times) and fun(char a='*', int times)
- ✓ Conflict between fun(char a='*', int times) and fun (int times)

fun('+', 60) என்னும் அழைப்புக் கூற்றை fun (char a, int times) மற்றும் fun (char a='*', int times) ஆகிய இரண்டு செயற்கூறுகளுடனும் பொருத்தமுடியும்.

வேறுபாடான அளபுருப் பட்டியல்கள் கொண்ட செயற்கூறுகளின் மூலமாகவோ ஒரு செயற்கூறினுக்குப் பணிமிகுக்க வேண்டும். அதாவது, அளபுருப் பட்டியல், அளபுருக்களின் எண்ணிக்கை அல்லது தரவினங்கள் அடிப்படையில் வேறுபட்டிருக்க வேண்டும்.

16.3 செயற்குறிப் பணிமிகுப்பு (Operator Overloading)

'செயற்குறி பணிமிகுப்பு' என்பது +,++,-,--,+=,-=,* <,>. போன்ற சி++ செயற்குறிகளுக்குக் கூடுதலான செயல்பாட்டினை வரையறுப்பதைக் குறிக்கிறது.

sum = num 1+ num2;

என்னும் கூற்று, இரண்டு எண்களைக் (int/float/double) கூட்டி, விடையை sum என்னும் மாறியில் இருத்தும் பணிக்காக கூற்றாகக் கருதிக் கொள்ளப்படும். இப்போது, கீழ்க்காணும் கூற்றை நோக்குக.

name = first_name + last_name;

இதில் name, first_name and last_name ஆகியவை மூன்றும் சரங்கள் (char இன அணிகள்) ஆகும். சி++ மொழியில் இவ்வாறு + செயற்குறி மூலம் இரண்டு சரங்களை ஒன்றிணைக்க முடியுமா? இரண்டு சரங்களை இணைக்க + செயற்குறியைப் பயன்படுத்த இயலாது என நிரல் பெயர்ப்பி பிழை சுட்டும். இரண்டு சரங்களை இணைக்க, பயனர் கட்டாயமாக strcat() செயற்கூறினையே பயன்படுத்த வேண்டும். எண் இனத்துக்குப் பயன்படுத்துவது போலவே சரங்களுக்கும் + செயற்குறியைப் பயன்படுத்த முடிந்தால் மிகவும் எளிதாக இருக்கும் அல்லவா? **செயற்குறிப் பணிமிகுப்பு** நுட்பம் மூலம் + செயற்குறியின் செயல்பாட்டைச் சரங்களுக்கும் நீட்டிக்க முடியும்.

கீழேயுள்ள எடுத்துக்காட்டைக் காண்க

```
//Program-16.2–OPERATOR OVERLOADING
 #include <iostream.h>
 #include <conio.h>
 #include <string.h>

 class strings
 {
    char s[10];
    public :
    strings()
    {
              s[0] = '\0';
    }
```

```
void main()
{
clrscr();
strings s1("test"),s2(" run\0");
char *concatstr ;
concatstr = s1 + s2;
cout << "\nConcatenated string ..."
     << concatstr;
getch();
}
```

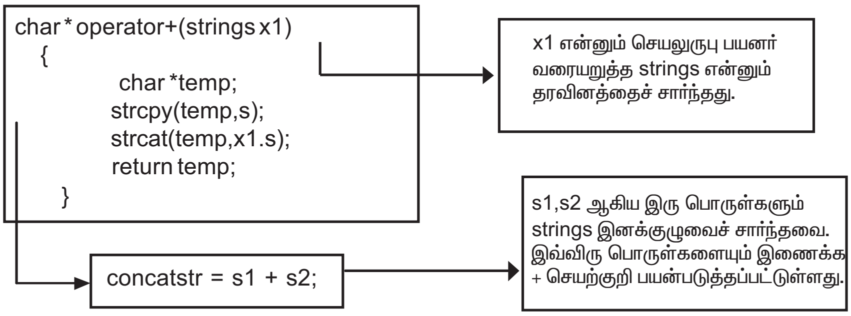
```
   strings(char *c)
   {
            strcpy(s,c);
   }

     char*operator+(strings x1)
      {
            char*temp;
            strcpy(temp,s);
            strcat(temp,x1.s);
            return temp;
   }
};
```

concatstr = s1 + s2 ;

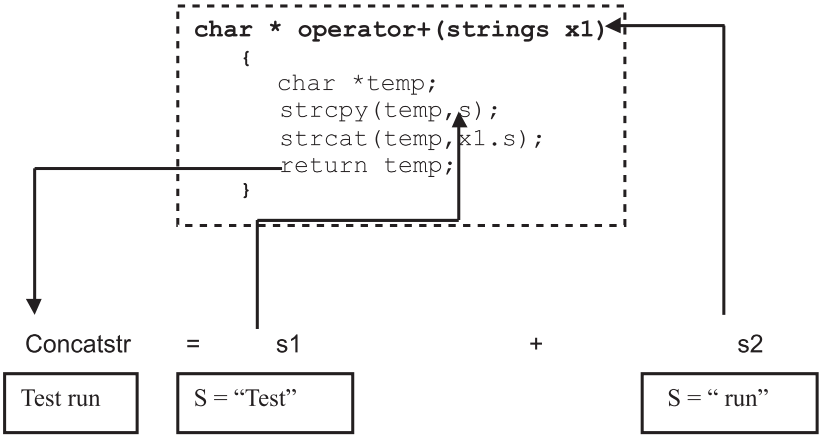
என்னும் கூற்று இரண்டு சரங்களை ஒன்றிணைக்கும்.

char * operator + (strings x1)

என்னும் உறுப்புச் செயற்கூறு மூலமாக, + செயற்குறிக்கு கூடுதலான பணி வரையறுக்கப்பட்டுள்ளது. இச்செயற்கூறு பயனர் வரையறுத்த x1 என்னும் தரவினைப் பொருளைச் செயலுருபாக ஏற்கிறது. இதனை இவ்வாறு விளக்கலாம்.



படம் **16.1** மாறிகளுக்கும் அவற்றின் மதிப்புகளுக்கும் இடையேயான தொடர்பினை விளக்குகிறது



படம் 16.1 மாறிகளுக்கும் மதிப்புகளுக்கும் உள்ள தொடர்பு

‘செயற்குறிப் பணிமிகுப்பு’ இவற்றையெல்லாம் வழங்குகிறது:

- ✓ +, *, -, ++, - -, >, <, += • மற்றும் இதுபோன்ற சி++ மொழியின் அடிப்படையான செயற்குறிகளுக்குப் புதிய வரையறைகளை வழங்குகிறது
- ✓ பயனர் வரையறுக்கும் தரவினங்களுக்கும் பணிமிகுத்த செயற்குறிகளை வரையறுக்க முடியும்.
- ✓ செயற்குறிகளுக்கான செயற்கூறுகள் உறுப்புச் செயற்கூறுகளாக இருக்க வேண்டும். அல்லது நட்புச் செயற்கூறுகளாக (friend functions) இருக்க வேண்டும். (நட்புச் செயற்கூறுகள் இந்த நூலின் வரையெல்லைக்கு அப்பாற்பட்டதாகும்).
- ✓ ஒரு செயற்குறிக்குத் தரப்படும் புதிய வரையறை அச்செயற்குறியின் மூல வரையறுப்பை மாற்ற முடியாது. எடுத்துக்காட்டாக, மேற்கண்ட நிரலில் + செயற்குறி இரண்டு சரங்களை ஒன்றிணைக்கப் பயன்படுத்தப்பட்டுள்ளது. அதே நிரலில் + செயற்குறியை வழக்கம் போல இரண்டு எண்களைக் கூட்டும் பணிக்கும் பயன்படுத்த முடியும். நிரல்பெயர்ப்பி, பயனர் வரையறுத்த செயற்குறியின் கட்டளைகளை ஓர் அழைப்புக் கூற்றைப் போலவே செயல்படுத்துகிறது. அதாவது ,

  ```
  cout << 5 + 10;
  ```

  என்னும் கட்டளை 15 எனத் திரையில் காட்டும் (+ன் மூல வரையறை செயல்படுத்தப்பட்டுள்ளது).

  ```
  concatstr=s1 + s2;
  ```

  என்னும் கட்டளை, char*operator+ (strings x1) என்னும் உறுப்புச் செயற்கூறினைச் செயல்படுத்தும், காரணம், இதில் + செயற்குறியின் இருபுறமும் இடம் பெற்றுள்ள செயலேற்பிகள் s1 , s2 ஆகிய இரண்டும் strings இனக்குழுவின் பொருள்களாகும்.

செயற்குறிப் பணிமிகுப்பில் நடைபெறும் செயல்பாடுகள்:

- ✓ ஒரு புதிய தரவினத்தை வரையறுக்கும் இனக்குழுவை உருவாக்கி அத்தரவினப் பொருள்களைப் பணிமிகுத்த செயற்குறியில் பயன்படுத்திக் கொள்ள வேண்டும்.
- ✓ operator <symbol>() செயற்குறியை இனக்குழுவின் public பகுதியில் அறிவிக்க வேண்டும்.
- ✓ தேவையான செயல்பாடுகளை நிறைவேற்றிக் கொள்ளும் வகையில் செயற்குறிக்கான செயற்கூறினை வரையறுக்க வேண்டும்.

பயனர் வரையறுத்த தரவினப் பொருள்கள் செயற்குறிகளில் எளிமையாகக் கையாளப்பட்டிருப்பதைக் கீழேயுள்ள எடுத்துக்காட்டு விளக்குகிறது. ஓர் இனக் குழுவில் உள்ள தரவு உறுப்புகளின் எதிர்ம மதிப்புகளைக் (negative value) கண்டறிய – செயற்குறியைப் பயன்படுத்தும் முறையை program- 16.3 விளக்குகிறது.

```
// Program - 16.3
#include <iostream.h>
#include <conio.h>

class negative
{
  int i;
  public :
  void accept()
  {
          cout << "\nEnter a number ...";
           cin >> i;
  }
  void display()
  {
           cout << "\nNumber ..."<<i;
  }
```

```
void operator-()
 {
        i = -i;
 }
};

void main()
{
        clrscr();
        negative n1, n2 ;
        n2.accept();
        -n1;
        n2.display();
        getch();
}
```

sum = - num1;

என்பதில், சாதாரண மாறிகளுக்கு – செயற்குறி செயல்படுவது போல, இனக் குழுவின் தரவு உறுப்புகளை void operator–() என்னும் செயற்கூறு எதிர்மம் (Negative)ஆக்குகிறது.

கீழேயுள்ள நிரலை நோக்குக. வினாக்களுக்கு விடை தருக:

```
// Program – 16.4
# include <iostream.h>
# include <conio.h>
class distance
{
        int feet,inches;
        public :
void distance_assign(int f, int i)
   {
        feet = f;
        inches = i;
   }
   void display()
   {
```

```
  cout << "\nFeet     : " << feet << "\tInches : " << inches;
    }
 distance operator+(distance d2)
   {
          distance d3;
          d3.feet = feet + d2.feet;
          d3.inches = (inches + d2.inches) % 12;
          d3.feet += (inches + d2.inches)/12;
          return d3;
    }
 };
 void main()
 {
          clrscr();
          distance dist_1,dist_2;
          dist_1.distance_assign(12,11);
          dist_2.distance_assign(24,1);
          distance dist_3 = dist_1 + dist_2;
          dist_1.display();
          dist_2.display();
          dist_3.display();
          getch();
 }
```

1. பணிமிகுக்கப்பட்ட செயற்குறிகளை அடையாளம் காண்க.
2. பணிமிகுக்கப்பட்ட உறுப்புச் செயற்கூறின் முன்வடிவை எழுதிக் காட்டுக.
3. பணிமிகுக்கப்பட்ட செயற்குறியில் பயன்படுத்தப்பட்டுள்ள செயலேற்பிகளின் இனம் யாது ?
4. பணிமிகுக்கப்பட்ட உறுப்புச் செயற்கூறினை இயக்கும் கூற்றினை எழுதிக் காட்டுக.

பணிமிகுக்கப்பட்ட +=, −= ஆகிய செயற்கூறுகளை program –16.5 விளக்குகிறது.

```
//Program-16.5
// operator overloading
# include <iostream.h>
# include <conio.h>
# include <string.h>
class library_book
{
     char name[25];
     int code,stock;
    public :
```

```
 void book_assign(char n[15],int c,int s)
 {
        strcpy(name,n);
        code = c;
        stock = s;
 }
 void display()
 {
       cout << "\n      Book details ....";
       cout << "\n      1. Name       ....." << name;
       cout << "\n      2. Code       ....." << code;
       cout << "\n      3. Stock      ....." << stock;
 }
 void operator +=(int x)
 {
       stock += x;
 }
       void operator -=(int x)
 {
      stock -= x;
 }
};
```

```
class library_cdrom
{
        char name[25];
        int code,stock;
        public :
        void cdrom_assign(char n[15],int c,int s)
        {
                strcpy(name,n);
                code = c;
                stock = s;
        }

        void display()
        {
                cout << "\n CD ROM details ....";
                cout << "\n 1. Name  ....." << name;
```

```
        cout << "\n 2. Code   ....." << code;
                cout << "\n 3. Stock    ....." << stock;
        }
        void operator +=(int x)
        {
                stock += x;
        }

        void operator -=(int x)
        {
                stock -= x;
        }
};

void main()
{
        library_book book;
        library_cdrom cdrom;

        book.book_assign("Half Blood Prince",101,55);
        cdrom.cdrom_assign("Know your Basics",201,50);

        char choice,borrow;

do
        {
        cout << "\nBook,cdrom,exit<b/c/e> ...";
        cin >> choice;
        if (choice != 'e')
                {
                        cout << "\nBorrow/Return <b/r> ...";
                        cin >> borrow;
                }
switch (choice)
         {
     case 'b' :
             switch (borrow)
         {
             case 'b' : book += 1;break;
             case 'r' : book -= 1;break;
         }
             book.display(); break;
    case 'c' :
            switch (borrow)
        {
            case 'b' : cdrom += 1;break;
            case 'r' : cdrom -= 1;break;
```

```
        }
           cdrom.display();
           break;

   case 'e' : cout << "\nTerminating ..";
                        break;
  }
  } while (choice != 'e');
  getch();
}
```

நூலகத்திலுள்ள இருப்புகள், வழக்கமான += , −= ஆகிய செயற்குறிகளைப் பயன்படுத்தி வழக்கமான பாணியில் மிக எளிதாக மிகுக்கப்படுவதையும், குறைக்கப்படுவதையும் கவனித்தீர்களா ?

```
book     += 1;
book     -= 1;
cdrom    += 1;
cdrom    -= 1;
```

> ஒரு செயற்குறிக்குப் புதிய பொருளை வழங்கும் செயல்நுட்பமே செயற்குறிப் பணிமிகுப்பு என்றழைக்கப்படுகிறது.

செயற்குறிப் பணி மிகுப்புக்கான விதிமுறைகள்

செயற்குறிகளின் பணிமிகுப்புக்குச் சில கட்டுப்பாடுகளும் வரம்புகளும் உள்ளன, அவை

- ✓ ஏற்கெனவே இருக்கும் செயற்குறிகளுக்கு மட்டுமே பணிமிகுக்க முடியும். புதிய செயற்குறிகளை உருவாக்க முடியாது.
- ✓ பணிமிகுக்கப்பட்ட செயற்குறி ஏற்கின்ற செயலேற்பிகளுள் ஒன்று மட்டுமாவது பயனர் வரையறுத்த தரவினமாக இருக்க வேண்டும்.
- ✓ ஒரு செயற்குறியின் அடிப்படை வரையறையை மாற்றியமைக்க முடியாது. அதாவது, ஒரு செயற்குறியின் செயல்பாட்டை மறுவரையறை செய்ய முடியாது. ஒரு செயற்குறிக்கு கூடுதலான பணிகளை வரையறுக்க முடியும்.
- ✓ பணிமிகுக்கப்பட்ட செயற்குறிகள் அவற்றின் செயலேற்பிகளைப் பொறுத்தமட்டில் அடிப்படைச் செயற்குறிகளைப் போன்றே செயல்படுகின்றன.
- ✓ இருமச் செயற்குறிகளை (Binary operators) பணிமிகுக்கும் போது, அச்செயற்குறியின் இடப்பக்கம் அமையும் பொருள், அது வரையறுக்கப்பட்டுள்ள இனக்குழுவின் பொருளாக இருக்க வேண்டும்.
- ✓ உறுப்புச் செயற்கூறு மூலம் இருமச் செயற்குறியைப் பணிமிகுக்கும் போது, ஒரேயொரு வெளிப்படைச் செயலுருபை மட்டுமே ஏற்கும்.

பயிற்சி வினாக்கள்

I. கீழேயுள்ள பணிகளை நிறைவேற்றச் செயற்கூறு பணிமிகுப்பைப் பயன்படுத்தும் ஒரு நிரலை எழுதுக:

அ) இரண்டு முழு எண்களில் பெரிய எண்ணைக் கண்டறியவும்

ஆ) மூன்று முழு எண்களில் பெரிய எண்ணைக் கண்டறியவும்

விடை : செயற்கூறு முன்வடிவுகள் max (int , int) and max(int , int, int)

II. கீழ்க்காணும் பணிகளுக்குப் பணிமிகுப்பு நுட்பத்தைப் பயன்படுத்திச் செயற்கூறுகளை எழுதவும் :

அ) float இன மாறியின் மதிப்பை ஒன்று மிகுக்க

ஆ) char இன மாறியின் மதிப்பை ஒன்று மிகுக்க

விடை : செயற்கூறு முன்வடிவுகள் – float incr (float) , char incr (char)

III. செயற்கூறு பணிமிகுப்பு மூலம் கீழ்க்காணும் பணிகளை நிறைவேற்ற சி++ மொழியில் ஒரு நிரல் எழுதுக :

அ) x,y ஆகியவை int எனில் , x^y மதிப்பைக் கண்டறியவும்.

ஆ) x,y ஆகிய இரண்டும் float எனில், x^y மதிப்பைக் கண்டறியவும்.

விடை : செயற்கூறு முன்வடிவுகள் – int power(int,int),float power(float,float);

IV. செயற்குறிப் பணிமிகுப்பின் ஆதாயங்கள் யாவை ?

V. பணிமிகுக்கப்பட்ட செயற்குறியை வரையறுப்பதில் பின்பற்ற வேண்டிய படிநிலைகளைப் பட்டியலிடுக.

VI. பணிமிகுப்புச் செய்ய முடியாத செயற்குறிகளைப் பட்டியலிடுக.

VII. complex_numbers என்னும் இனக்குழுவின் இரண்டு பொருள்களைக் கூட்டிச் சொல்ல நிரல் எழுதுக. ஒரு கலப்பு எண் என்பது மெய்ப்பகுதி, கற்பனைப் பகுதி என்னும் இரண்டு தரவு உறுப்புகளைக் கொண்டிருக்கும். கீழேயுள்ள வரையறையை நிறைவு செய்து, complex_numbers இனத்திலுள்ள c1, c2 என்னும் இரண்டு பொருள்களைக் கூட்டிச் சொல்ல main() செயல்கூறினை எழுதிக் காட்டுக.

```
Class complex_numbers
{
float x;
float y;
public :
void assign_data(float real, float imaginary);
void display_Data();
complex_numbers operator +(complex_numbers n1);
}
```

அதிகாரம் 17

ஆக்கிகளும் அழிப்பிகளும்

(Constructors and Destructors)

17.1 முன்னுரை

ஓர் இனக்குழுவின் **சான்றுரு** (instance) பயன்பாட்டுக்கு வரும்போது, **ஆக்கி** (constructor) எனப்படும் சிறப்புச் செயற்கூறு இயக்கப்படுகிறது. ஆக்கி செயற்கூறு இனக்குழுப் பொருளைத்(class object) தொடங்கிவைக்கிறது அதாவது, அதன் உறுப்புகளில் தொடக்க மதிப்பு இருத்துகிறது. ஓர் இனக்குழுப் பொருளின் பயன்பாடு முடிவுக்கு வரும்போது **அழிப்பி** (destructor) எனப்படும் சிறப்புச் செயற்கூறு இயக்கப்படுகிறது. ஆக்கி, அழிப்பி ஆகிய இரண்டு செயற்கூறுகளுமே இனக்குழுவின் பெயரையே கொண்டுள்ளன. இரண்டு செயற்கூறுகளுமே எந்த மதிப்பையும் திருப்பி அனுப்புவதில்லை. ஏந்தத் தரவினத்தோடும் தொடர்புடையவை அல்ல.

17.2 ஆக்கிகள் (constructors)

```
// Program - 17.1
// to determine constructors and destructors
#include<iostream.h>
#include<conio.h>
class simple
{
  private:
          int a,b;
public:
simple()
{
  a= 0 ;
  b= 0;
  cout<< "\n Constructor of class-simple ";
}
   ~simple()
{
   cout<<"\n Destructor of class – simple .. ";
}
   void getdata()
{
   cout<<"\n Enter values for a and b... ";
   cin>>a>>b;
}
   void putdata()
{
   cout<<"\nThe two integers .. "<<a<<'\t'<< b;
   cout<<"\n The sum of the variables .. "<< a+b;
}
};
   void main()
{
   simple s;
   s.getdata();
   s.putdata();
}
```

மேற்கண்ட நிரலைச் செயல்படுத்துகையில் s என்னும் பொருள் உருவாக்கப்படும் போது ~simple என்னும் ஆக்கி, தானாகவே இயக்கப்படுகிறது. s–ன் பயன்பாடு முடியும்போது, அதாவது நிரலின் செயல்பாடு முடிவுக்கு வரும் போது ~simple() என்னும் அழிப்பி தானாகவே இயக்கப்படுகிறது.

இந்த நிரலின் வெளியீடு இவ்வாறு இருக்கும்

Constructor of class - simple ..

Enter values for a & b…5 6

The two integers….. 5 6

The sum of the variables….. 11

Destructor of class - simple …

17.3 ஆக்கியின் செயல்பாடுகள்

1. இனக்குழுப் பொருளின் உறுப்புகளில் தொடக்க மதிப்பு இருத்துகிறது

2. பொருளுக்கு நினைவகத்தில் இடம் ஒதுக்குகிறது.

17.4 ஆக்கியின் பணிமிகுப்பு (Constructor overloading)

ஆக்கிகள் இனக்குழுவின் சிறப்புச் செயற்கூறுகள் என்பதால், செயற்கூறு பணிமிகுப்பு (function Overloading) ஆக்கிகளுக்கும் பொருந்தும். program 17.2 நிரல் ஆக்கிப் பணிமிகுப்பை விளக்குகிறது

add() என்னும் ஆக்கி **அளபுருக்கள் இல்லாத ஆக்கி** (non-parameterized constructor) ஆகும். இது **தானமைவு ஆக்கி** (default constructor) எனப்படுகிறது. வழக்கமாக, தானமைவு ஆக்கிகள் **நிரல்பெயர்ப்பி உருவாக்கும் ஆக்கிகள்** (complier generated constructors) எனக் குறிப்பிடப்படுவதுண்டு. அதாவது பயனர் வரையறுத்த ஆக்கிகள் எதுவும் இல்லாத போது கணிப்பொறியே இத்தகைய ஆக்கியை உருவாக்கிக் கொள்கிறது. அளபுருக்கள் இல்லாமல் ஒரு பொருள் அறிவிக்கப்படும்போது, அளபுருக்கள் இல்லாத ஆக்கி இயக்கப்படுகிறது.

```
// Program - 17.2
// To demonstrate constructor overloading
# include<iostream.h>
#include<conio.h>
class add
{
        int num1, num2, sum;
        public:
        add()
        {
                cout<<"\n Constructor without parameters.. ";
                num1= 0;
                num2= 0;
                sum = 0;
        }
```

```
add ( int s1, int s2 )
          {
                    cout<<"\n Parameterized constructor... ";
                    num1= s1;
                    num2=s2;
                    sum=NULL;
          }
          add (add &a)
          {
                    cout<<"\n Copy Constructor ... ";
                    num1= a.num1;
                    num2=a.num2;
                    sum = NULL;
          }
          void getdata()
          {
                    cout<<"Enter data ... ";
                    cin>>num1>>num2;
          }
          void addition()
          {
          sum=num1+num2;
          }
          void putdata()
          {
          cout<<"\n The numbers are..";
          cout<<num1<<'\t'<<num2;
          cout<<"\n The sum of the numbers are.. "<< sum;
          }
};
void main()
{
          add a, b (10, 20) , c(b);
          a.getdata();
          a.addition();
          b.addition();
          c.addition();
          cout<<"\n Object a :  ";
          a.putdata();
          cout<<"\n Object b :  ";
          b.putdata();
          cout<<"\n Object c.. ";
          c.putdata();
}
```

வெளியீடு

Constructor without parameters….

Parameterized Constructor...

Copy Constructors…

Enter data .. 5 6

Object a:

The numbers are 5 6

The sum of the numbers are ….. 11

Object b:

The numbers are 10 20

The sum of the numbers are … 30

Object c:

The numbers are 10 20

The sum of the numbers are ….. 30

add (int s1, int s2) என்னும் ஆக்கி **அளபுரு ஏற்கும் ஆக்கி** எனப்படுகிறது. இந்த ஆக்கியைச் செயல்படுத்த, இரண்டு int மாறிலிகள் அல்லது மாறிகளுடன் பொருளை அறிவிக்க வேண்டும்.

குறிப்பு :– char, float, double அளபுருக்களையும் ஏற்கும். அவை உள்ளுறை இனமாற்றத்தின் காரணமாக int தரவினத்தோடு பொருந்தும் என்பதால் ஏற்றுக் கொள்ளப்படக் கூடியவையே.

எடுத்துக்காட்டு: add a (10, 60) / add a (ivar, ivar)

add (add & a) என்னும் ஆக்கி **நகல் ஆக்கி** (copy constructor) எனப்படுகிறது. நகல் ஆக்கி கீழ்க்காணும் சூழல்களில் இயக்கப்படும்.

1) ஏதேனும் ஓர் உறுப்புச் செயற்கூறினுக்கு ஒரு பொருளை அளபுருவாக அனுப்பிவைக்கும்போது

   எடுத்துக்காட்டு void add::putdata(add x);

2) ஓர் உறுப்புச் செயற்கூறு ஒரு பொருளைத் திருப்பியனுப்பும் போது,

   எடுத்துக்காட்டு, add getdata();

3) ஓர் பொருள் குறிப்புவகை அளபுருவாக ஆக்கிக்கு அனுப்பி வைக்கப்படும்போது,

எடுத்துக்காட்டு add a, b(a);

கீழேயுள்ள நிரல் (program 17.3) நகல் ஆக்கி எப்போதெல்லாம் இயக்கப்படுகிறது என்பதை விளக்குகிறது.

```
// Program – 17.3
// To demonstrate constructor overloading
# include<iostream.h>
#include<conio.h>
class add
{
        int num1, num2, sum;
        public:
        add()
        {
                cout<<"\n Constructor without parameters.. ";
                num1= 0;
                num2= 0;
                sum  = 0;
        }
        add ( int s1, int s2 )
        {
                cout<<"\n Parameterized constructor... ";
                num1= s1;
                num2=s2;
                sum=NULL;
        }
        add (add &a)
        {
                cout<<"\n Copy Constructor ... ";
                num1= a.num1;
                num2=a.num2;
                sum = NULL;
        }
        void getdata()
        {
                cout<<"Enter data ... ";
                cin>>num1>>num2;
        }
        void addition(add b)
        {
                sum=num1+ num2 +b.num1 + b.num2;
        }
        add addition()
        {
                add a(5,6);
                sum = num1 + num2 +a.num1 +a.num2;
        }
        void putdata()
        {
                cout<<"\n The numbers are..";
                cout<<num1<<'\t'<<num2;
                cout<<"\n The sum of the numbers are.. "<< sum;
        }
};
```

```
void main()
{
	clrscr();
	add a, b (10, 20) , c(b);
	a.getdata();
	a.addition(b);
	b = c.addition();
	c.addition();
	cout<<"\n Object a :     ";
	a.putdata();
	cout<<"\n Object b :     ";
	b.putdata();
	cout<<"\n Object c.. ";
	c.putdata();
}
```

ouput of the above program

```
Constructor without parameters..
Parameterized constructor...
Copy Constructor ... Enter data ... 2        3
Copy Constructor ...
Parameterized constructor...
Parameterized constructor...
Object a :
The numbers are..2          3
The sum of the numbers are.. 35
Object b :
The numbers are..0          1494
The sum of the numbers are.. 0
Object c :
The numbers are..10         20
The sum of the numbers are.. 41
```

நகல் ஆக்கி எத்தனை முறை இயக்கப்படுகிறது என்பதைக் கவனித்தீர்களா ?

program 17.3 நிரலில் கீழ்க்காணும் கூற்றுகள் நகல் ஆக்கியை இயக்குகின்றன.

add c (b);// b என்னும் பொருள் செயலுருபாக அனுப்பி வைக்கப்படுகிறது
a.addition(b);// b என்னும் பொருள் addition() என்னும் உறுப்புச் செயற்கூறினுக்கு செயலுருபாக அனுப்பி வைக்கப்படுகிறது.
b=c.addition();// addition() என்னும் உறுப்புச் செயற்கூறு ஒரு பொருளைத் திருப்பியனுப்புகிறது

மேற்கண்ட எடுத்துக்காட்டில் addition() என்னும் செயற்கூறின் பணி மிகுக்கப்பட்டுள்ளது (overloaded). நிரலினுள் எந்த இடத்தில் அறிவிக்கப்படும் செயற்கூறுகளும் பணிமிகுக்கப்பட முடியும் என்பதறிக.

17.5 ஆக்கி வரையறுப்பு மற்றும் பயன்படுத்தலுக்கான விதிமுறைகள் :

1. ஆக்கியின் பெயர் இனக்குழுவின் பெயராகவே இருக்க வேண்டும்.
2. ஆக்கி, அளபுருக்களின் பட்டியலைக் கொண்டிருக்க முடியும்
3. ஆக்கிச் செயற்கூறு, பணிமிகுக்கப்பட முடியும்.
4. ஆக்கி எதையும் பயனர் வரையறுக்காத போது, நிரல்பெயர்ப்பி ஓர் ஆக்கியை உருவாக்கிக் கொள்ளும்.
5. ஓர் இனக்குழுப் பொருள் உருவாக்கப்படும் போது, ஆக்கியானது தானாகவே இயக்கப்படும். அழைக்க வேண்டியதில்லை.

17.6 அழிப்பிகள் (Destructors)

அழிப்பி என்பது, ஒரு பொருளை உருவாக்கும் போது, ஆக்கியால் பொருளுக்கென ஒதுக்கப்படும் நினைவகப் பகுதியை விடுவிக்கும் ஒரு செயற்கூறாகும். இதுவும் இனக்குழுவின் பெயரையே கொண்டிருக்கும். ஆனால் ~ என்னும் குறியைப் பெயரின் முன்னொட்டாகக் கொண்டிருக்கும்.

எடுத்துக்காட்டு :

```
class simple
    {
        ____
        ____
        public :
        ~simple()
        {
        ......................
        }
    }
```

17.7 அழிப்பி வரையறுப்பு மற்றும் பயன்படுத்தலுக்கான விதிமுறைகள்

1. அழிப்பியின் பெயரானது ~ என்ற முன்னொட்டுக் குறியுடன் கூடிய இனக்குழுவின் பெயரையே கொண்டிருக்கும்.

2. அழிப்பி, செயலுருபுகளை ஏற்காது.

3. அழிப்பி, எந்த மதிப்பையும் திருப்பி அனுப்பாது.

4. அழிப்பி பணிமிகுக்கப்பட முடியாது. அதாவது, ஓர் இனக்குழுவில் ஓர் அழிப்பி மட்டுமே இருக்க முடியும்.

5. பயனர், அழிப்பியை வரையறுக்காத போது, நிரல்பெயர்ப்பி ஓர் அழிப்பியை உருவாக்கிக் கொள்ளும்.

6. நிரலில் உருவாக்கப்பட்ட ஓர் இனக்குழுப் பொருளின் பயன்பாடு முடிவுக்கு வரும்போது, அழிப்பி தானாகவே இயக்கப்படும். அழைக்க வேண்டியதில்லை.

பயிற்சி வினாக்கள்

I. கீழ்க்காணும் அட்டவணையை நிறைவு செய்க

வினாக்கள்	ஆக்கி	அழிப்பி
1. எந்த வரையெல்லைக்குள் (scope) அறிவிக்கப்பட வேண்டும் **?**		
2. பணிமிகுப்பு சாத்தியமா **?**		
3. எப்போதுஇயக்கப்படுகிறது **?** ஒர் இனக் குழுப்பொருளானது.............		
4. செயலுருபுகளை ஏற்குமா **?**		

II. கீழ்க்காணும் நிரல்பகுதிகள் பிழைசுட்டுவது ஏன்?

```
Class simple
{
  private :
   int x;
  simple()
 { x = 5; }
};
```

```
Class simple
{
  private :
   int x;
 public :
   simple(int y)
 { x = y; }
};
void main()
{
  simple s;
}
```

```
Class simple
{
  private :
   int x;
 public :
  simple(int y)
 { x = y; }

  simple(int z = 5)
{
    x = z;
}
};
void main()
{
  simple s(6);
}
```

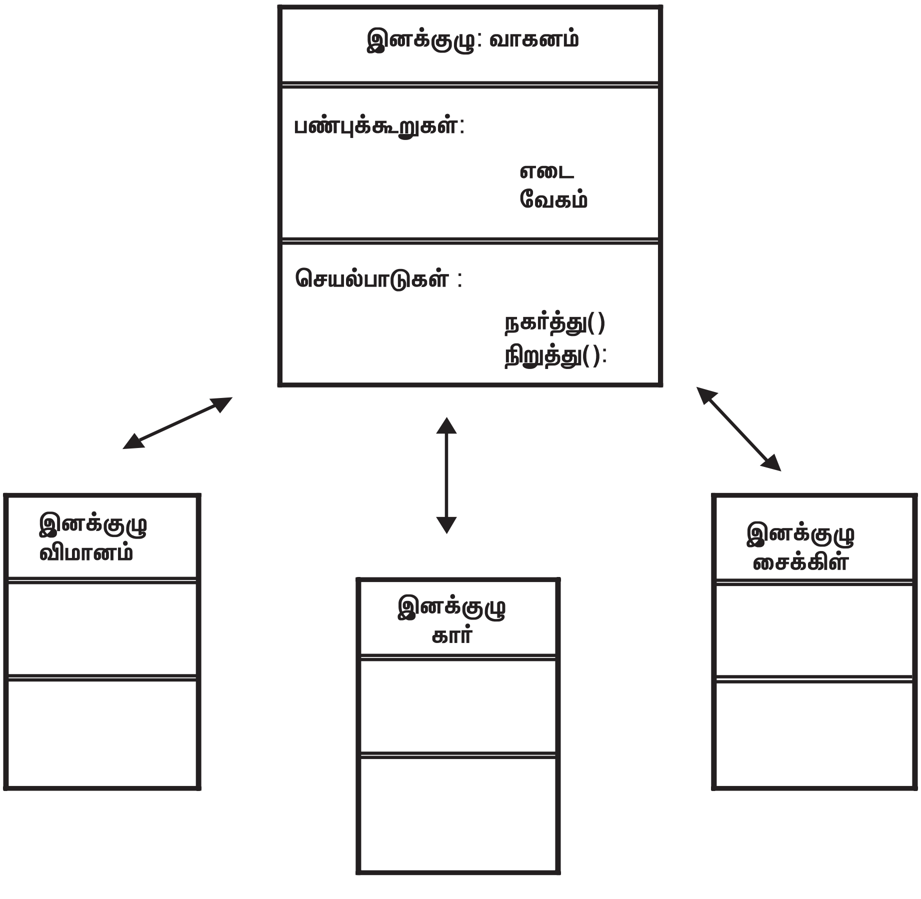
அதிகாரம் 18

மரபுரிமம்

(INHERITANCE)

18.1 முன்னுரை

ஒரு பொருள்நோக்கு நிரலாக்க மொழியின் சக்தி வாய்ந்த பண்புக்கூறு **மரபுரிமம்** (Inheritance) ஆகும். மரபுரிமம் என்பது ஏற்கெனவே இருக்கும் இனக்குழுக்களின் அடிப்படையில் புதிய இனக்குழுக்களை உருவாக்கும் செயல் முறை ஆகும். இவை முறையே **அடிப்படை இனக்குழுக்கள்** (Base classes) எனவும், **தருவிக்கப்பட்ட இனக்குழுக்கள்** (derived classes) எனவும் அழைக்கப்படுகின்றன. தருவிக்கப்பட்ட இனக்குழு, அடிப்படை இனக்குழுவின் அனைத்துப் பண்புகளையும் மரபுரிமையாகப் பெற்று விடுகிறது. அதுமட்டுமின்றி, தருவிக்கப்பட்ட இனக்குழுவில் கூடுதலான பண்புக்கூறுகளையும் வழிமுறைகளையும் சேர்த்து அதன் செயலாற்றலை அதிகரிக்க முடியும். 'மரபுரிமம்' என்னும் சொல் நடைமுறை வாழ்க்கையில் நமக்கு மிகவும் அறிமுகமான சொல். குழந்தைகள் தமக்கே உரிய பண்பியல்புகளுடன் பெற்றோரின் பண்புகளை மரபு வழியாகப் பெறுகின்றன. அதுபோலவே, ஓர் இனக்குழுவும் அதன் அடிப்படை (parent) இனக்குழுவிடமிருந்து பண்புகளை மரபுரிமையாகப் பெறுகிறது.



படம்.18.1 மரபுரிமம்

18.2 மரபுரிமத்தின் பலன்கள்

மரபுரிமம் கீழ்க்காணும் பலன்களை அளிக்கிறது

1. நிரல் குறிமுறையின் மறுபயனாக்கம். (Reusability of Code);

   ஒரு நிறுவனத்தில் பல்வேறு பயன்பாட்டு மென்பொருள்கள் உருவாக்கப்படுகின்றன. ஒரு பயன்பாட்டுக்கென எழுதப்பட்ட குறிமுறைத் தொகுதியை, அதே செயலாக்கம் வேண்டப்படுகின்ற வேறொரு பயன்பாட்டில் பயன்படுத்திக் கொள்ள முடியும். இதனால் நிரல் உருவாக்கும் நேரம் வெகுவாகக் குறைகிறது.

2. நிரல் குறிமுறைப் பகிர்வு (Code sharing);

   அடிப்படை இனக்குழுவின் வழிமுறைகளை (methods) தருவிக்கப்பட்ட இனக்குழுக்கள் பகிர்ந்து கொள்ள முடியும்.

3. இடைமுகத்தின் முரண்பாடின்மை (Consistency of interface);

   மரபுரிமையாகப் பெறும் பண்புக் கூறுகளும் வழிமுறைகளும் அழைக்கும் வழிமுறைகளுக்கு ஒரே மாதியான இடைமுகத்தையே வழங்குகின்றன. படம் 18.1 ல் கண்டுள்ளபடி, **வாகனம்** என்னும் இனக்குழுவின் பண்புக்கூறுகளும், வழிமுறைகளும், **விமானம், கார், சைக்கிள்** ஆகிய மூன்று தருவிக்கப்பட்ட இனக்குழுக்களுக்கும் பொதுவாக அமைந்துள்ளன. இந்த மூன்று தருவிக்கப்பட்ட இனக்குழுக்களுமே முரண்பாடற்ற இடைமுகங்களைக் கொண்டுள்ளன எனலாம்.

18.3 தருவிக்கப்பட்ட இனக்குழுவும் அடிப்படை இனக்குழுவும்

ஓர் இனக்குழுவிலிருந்து பிற இனக்குழுக்கள் தருவிக்கப்படுமாயின் அதனை **அடிப்படை** இனக்குழு என்கிறோம். அடிப்படை இனக்குழுவின் உறுப்புகளைத் தருவிக்கப்பட்ட இனக்குழு மரபுரிமையாகப் பெறுகிறது.

தருவிக்கப்பட்ட இனக்குழுவை வரையறுக்கும் போது, கீழ்க்காணும் விதிமுறைகளைப் பின்பற்ற வேண்டும்.

அ) **class** என்னும் சிறப்புச் சொல் இருக்க வேண்டும்.

ஆ) class என்ற சொல்லை அடுத்து, தருவிக்கப்படும் இனக்குழுவின் பெயர் இடம் பெற வேண்டும்.

இ) ஒற்றை முக்காற்புள்ளி(**:**) இடம்பெற வேண்டும்

ஈ) private, Public அல்லது protected ஆகியவற்றுள் எத்தகைய அணுகியல்புடன் தருவிக்கப்படுகிறது எனக் குறிப்பிட வேண்டும்.

உ) அடிப்படை இனக்குழுவின் பெயரைகுறிப்பிட வேண்டும்.

ஊ) தருவிக்கப்படும் இனக்குழுவுக்குரிய கூடுதல் பண்புகளையும், வழிமுறைகளையும் வரையறுக்க வேண்டும்.

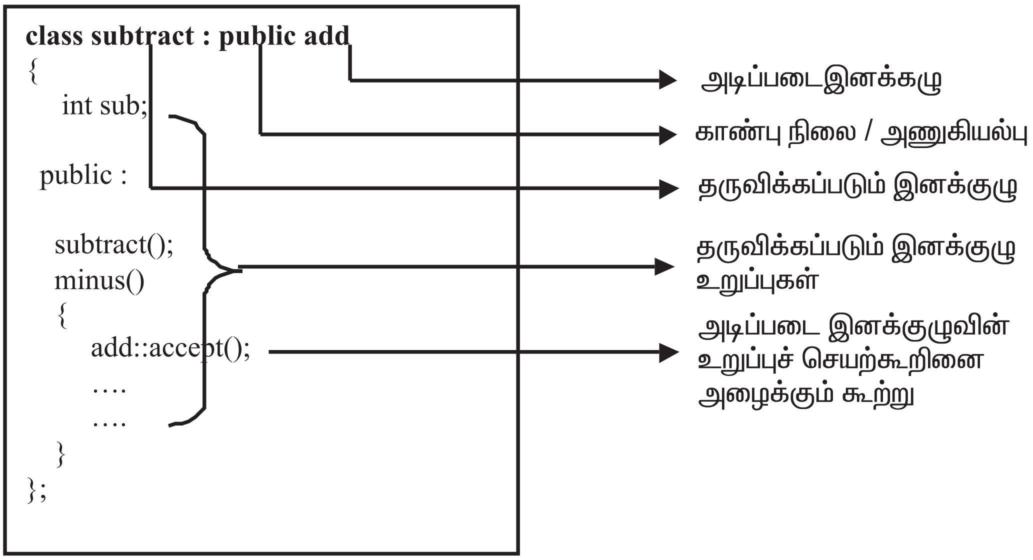
```
// Program - 18.1
#include< iostream.h>
#include<conio.h>

class add
{
        int sum;
        protected:
                int num1, num2;
        public:
        add()
        {
                num1= num2= sum=0;
                cout<<"\n Add constructor .. ";
        }
        accept()
        {
                cout<<"\n Enter two numbers .. ";
                cin>>num1>>num2;
        }
plus()
        {
                sum = num1 + num2;
                cout<<"\n The sum of two numbers is .. "<< sum;
        }
};
class subtract :public add
{
        int sub;
        public:
        subtract()
         {
           sub = 0;
           cout<<"\n Subtract constructor .. ";
         }
         minus()
         {
                add::accept();
                sub= num1-num2;
                cout<<"\n The difference of two numbers are ... "
                        << sub;
        }
 };
```

```
void main()
{
      subtract s;
      int choice = 0;
      cout<<"\n Enter your choice ";
      cout<<" \n1. Add..\n2. Subtract ..";
      cin>>choice;
      switch( choice )
      {
            case 1:
                  s.accept();
                  s.plus();
                  break;
            case 2:
                  s.minus();
                  break;
      }
}
```

program-18.1 ல் add என்பது அடிப்படை இனக்குழு subtract என்பது தருவிக்கப்பட்ட இனக்குழு. தருவிக்கப்பட்ட இனக்குழு இவ்வாறு வரையறுக்கப்பட வேண்டும்.

```
class <பெயர்> : <அணுகியல்பு> <அடிப்படை இனக்குழுவின் பெயர்>
{
      தருவிக்கப்படும் இனக்குழுவின் தரவு உறுப்புகள்;
      தருவிக்கப்படும் இனக்குழுவின் உறுப்புச் செயற்கூறுகள்;
}
```

program - 18.1 ல் substract என்னும் தருவிக்கப்பட்ட இனக்குழு இவ்வாறு வரையறுக்கப்பட்டுள்ளது.



18.4 காண்புநிலைப் பாங்கு/ அணுகியல்பு வரையறுப்பி

(Visibility Mode/Accessibility specifier)

மரபுரிமத்தின் குறிப்பிடத்தக்க கூறு என்னவெனில், தருவிக்கப்பட்ட இனக்குழுவின் பொருள்களும் உறுப்புகளும் அடிப்படை இனக்குழுவின் உறுப்புகளை எப்பொழுது, எப்படிப் பயன்படுத்திக் கொள்ள வேண்டும் என்பதுதான். இதுவே, **அணுகியல்பு**(accessibility) எனப்படுகிறது. public, private, protected என மூன்று **அணுகியல்பு** வரையறுப்பிகள் உள்ளன. அணுகியல்பு வரையறுப்பியை **காண்புநிலைப் பாங்கு** (visibility mode) என்றும் கூறலாம். முன்னியல்பான காண்புநிலைப் பாங்கு private ஆகும். தருவிக்கப்பட்ட இனக்குழுவில், அடிப்படை இனக்குழு உறுப்புகளின் வரையெல்லை மற்றும் அணுகியல்பை அட்வணை 18.1 விளக்குகின்றது.

அடிப்படை இனக் குழுவின் உறுப்புகள்	**தருவிக்கப்பட்ட இனக்குழு**		
	private	protected	public
private உறுப்புகள்	மரபுரிமையாகப் பெறப்படுவதில்லை. ஆனாலும் அவை தொடர்ந்து நிலவும்.	–	–
protected உறுப்புகள்	private அணுகியல்பு கொண்ட உறுப்புகளாக மரபுவழி பெறப்படுகிறது.	protected அணுகியல்பு கொண்ட உறுப்புகளாக மரபுவழி பெறப்படுகிறது.	protected அணுகியல்பு தக்க வைத்துக் கொள்ளப்படுகிறது.
public உறுப்புகள்	private அணுகியல்பு கொண்ட உறுப்புகளாக மரபுவழி பெறப்படுகிறது.	protected அணுகியல்பு கொண்ட உறுப்புகளாக மரபுவழி பெறப்படுகிறது.	public அணுகியல்பு கொண்ட உறுப்புகளாகவே பெறப்படுகிறது.

அட்டவணை 18.1 தருவிக்கப்பட்ட இனக்குழுவில் அடிப்படை இனக்குழு உறுப்புகளின் வரையெல்லையும் அணுகியல்பும்.

ஓர் அடிப்படை இனக்குழு private என்னும் அணுகியல்புடன் தருவிக்கப்படும் போது, அடிப்படை இனக்குழுவின் public மற்றும் protected உறுப்புகள், தருவிக்கப்படும் இனக்குழுவில் private உறுப்புகளாகவே நிலவுகின்றன.

ஓர் அடிப்படை இனக்குழு protected என்னும் அணுகியல்புடன் தருவிக்கப்படும் போது, அடிப்படை இனக்குழுவின் protected மற்றும் public உறுப்புகள், தருவிக்கப்படும் இனக்குழுவில் protected உறுப்புகளாக நிலவுகின்றன.

ஓர் அடிப்படை இனக்குழு public என்னும் அணுகியல்புடன் தருவிக்கப்படும் போது, அடிப்படை இனக்குழுவின் protected உறுப்புகள், தருவிக்கப்படும் இனக்குழுவில் protected உறுப்புகளாகவும், public உறுப்புகள் public உறுப்புகளாகவும் நிலவுகின்றன.

> அடிப்படை இனக்குழுக்கள் public, private, protected என எப்படித் தருவிக்கப்பட்டாலும் அடிப்படை இனக்குழுவின் private உறுப்புகள் தருவிக்கப்படும் இனக்குழுவில் மரபுரிமையாகப் பெறப்படுவதில்லை. தருவிக்கப்பட்ட இனக்குழுவில் அவை நிலவும். ஆனால் அவற்றை அணுக முடியாது.

program **18.1** add, substract ஆகிய இனக்குழுக்கள் இவ்வாறு அறிவிக்கப்பட்டுள்ளன.

```
class add
{
        private:
                int sum;
        protected :
                int num1, num2;
        public:
                add();
                accept();
                plus();
};
class subtract : private add
{
        int sub;
     public:
        subtract();
        minus();
};
```

subtract இனக்குழு மரபுவழிப் பெற்ற தரவு உறுப்புகளும், உறுப்புச் செயற்கூறுகளும்:

int num1 & num2 - subtract - ல் இவை private

accept() plus() - subtract - ல் இவை private

I

```
class subtract : private add
{
        int sub;
  public:
       subtract( );
       minus();
};
```

II

```
class subtract : protected add
{
        int sub;
  public:
       subtract();
       minus();
};
```

III

```
class subtract : public add
{
        int sub;
  public:
       subtract();
       minus();
};
```

அடிப்படை இனக்குழு உறுப்புகளின் அணுகியல்பு அட்டவணை 18.2ல் தரப்பட்டுள்ளன.

அடிப்படை இனக்குழுவின் ஆக்கிகள் (constructors) தருவிக்கப்பட்ட இனக்குழுவில் மரபுவழி பெறப்படுவதில்லை. ஆனால், தருவிக்கப்பட்ட இனக்குழுவில் ஒரு சான்றுரு (instance) உருவாக்கப்படும்போது முதலில் இயக்கப்படுகின்றன.

Class subtract	II	III
private:	private:	private:

sub
num1
num2
accept()
plus()
public:
subtract()
minus();
private mode
of inheritance
	sub
	protected:
	num1, num2
	accept();
	plus();
	public:
	subtract()
	minus()
	protected mode
	of inheritance
		sub
		protected:
		num1, num2
		public:
		accept();
		plus();
		subtract()
		minus();
		public mode
		of inheritance

அட்டவணை 18.2 அடிப்படை இனக்குழு உறுப்புகளின் அணுகியல்பு

program 18.1ல் அறிவிக்கப்பட்டுள்ள பொருள்களை நோக்குக. இந்த நிரலில் அடிப்படையில் கீழ்க்காணும் அட்டவணையை நிறைவு செய்க.

இனக்குழுவின் பொருள்கள்	**தரவு உறுப்புகள்**	**வழிமுறைகள்/ செயற்கூறுகள்**
add		
subtract		

அட்டவணை 18.3 பொருள்களின் தரவு / செயற்கூறு உறுப்புகள்

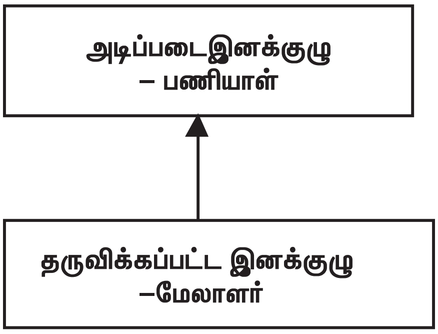
இயக்கப்படும் ஆக்கிகள் :................................... , ...

18.5 மரபுரிமத்தின் வகைகள்

தருவிக்கப்பட்ட இனக்குழுக்களை அடிப்படையாகக் கொண்டு புதிய இனக்குழுக்களைத் தருவிக்க முடியும். மரபுரிமத்தில் பலவகைகள் உள்ளன. ஒரு வழி மரபுரிமம், பலவழி மரபுரிமம், பலநிலை மரபுரிமம், கலப்பு மரிபுரிமம், படிமுறை மரபுரிமம்.

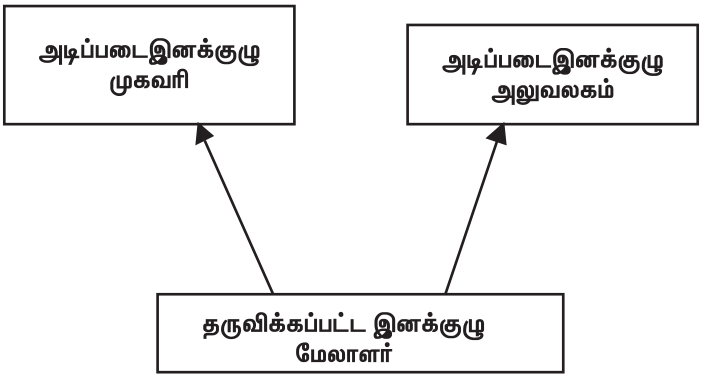
1) ஒருவழி மரபுரிமம் (Single Inheritance)

ஒரேயொரு இனக்குழுவை அடிப்படையாகக் கொண்டு தருவிக்கப்பட்ட இனக்குழுவை உருவாக்குவது ஒருவழி மரபுரிமம் ஆகும்.



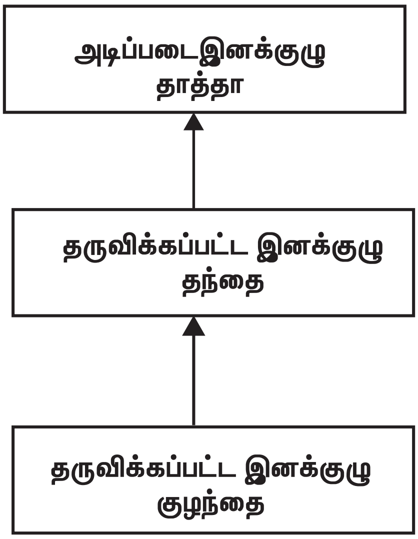
2) பல வழி மரபுரிமம் (Multiple Inheritance)

ஒன்றுக்கு மேற்பட்ட அடிப்படை இனக்குழுக்களிலிருந்து தருவிக்கப்பட்ட இனக்குழுவை உருவாக்குவது பலவழி மரபுரிமம் ஆகும்.



3) பலநிலை மரபுரிமம் (Multilevel Inheritance)

தருவிக்கப்பட்ட ஓர் இனக்குழுவை அடிப்படையாகக் கொண்டு இன்னொரு தருவிக்கப்பட்ட இனக்குழுவை உருவாக்குவது பலநிலை மரபுரிமம் ஆகும்.



Program 18.2 நிரலின் வெளியீடு என்னவாக இருக்கும் ?

```
// Program - 18.2
#include<iostream.h>
#include<conio.h>
class base
{
            public:
            base()
           {
              cout<<"\nConstructor of base class...";
           }
           ~base()
           {
              cout<<"\nDestructor of base class.... ";
           }
};
class derived:public base
{
           public :
           derived()
            {
               cout << "\nConstructor of derived ...";
            }
           ~derived()
           {
```

```
cout << "\nDestructor of derived ...";
           }
};
class derived2:public base
 {
   public :
   derived()
   {
       cout << "\nConstructor of derived2 ...";
   }
  ~derived()
{
      cout << "\nDestructor of derived2 ...";
 }
};
    void main()
{
    derived2 x;
}
```

வெளியீடு

Constructor of base class…

Constructor of derived ….

Constructor of derived2 …

Destructor of derived2…

Destructor of derived

Destructor of base class ..

✓ **ஆக்கிகள்** மரபுரிமம் பெற்ற இனக்குழுக்களின் வரிசையில் இயக்கப்படுகின்றன. அதாவது, அடிப்படை இனக்குழுவில் தொடங்கி இறுதியாகத் தருவிக்கப்பட்ட இனக்குழு வரையில், **அழிப்பிகள்** முன்பின் வரிசையில் இயக்கப்படும்.

பிற இனக்குழுக்களை தருவிப்பதற்காக மட்டுமே பயன்படும் இனக்குழுக்கள் அருவ இனக்குழுக்கள் அல்லது **கருத்தியல் இனக்குழுக்கள்(Abstract Classes)** எனப்படுகின்றன. அதாவது, கருத்தியல் இனக்குழுக்களில் பொருள்களை உருவாக்க முடியாது.

பயற்சி வினாக்கள்

1) கீழ்க்காணும் வரையறுப்புகளை அடிப்படையாகக் கொண்டு, அட்டவணை **4, 5, 6**, ஆகியவற்றை நிறைவு செய்க.

```
class node
{
        int x;
        void init();
        public:
        void read();
        protected:
        void get();
};
class type : public node
{
        int a;
        public:
        void func1();
        protected:
        int b;
        void getb();
}
class statement :private type
{
        int p;
        public:
        void func2();
        protected:
        void func3();
};
```

type இனக் குழுவின் உறுப்புகள்	**உறுப்புகளின் அணுகியல்பு**		
	private	protected	public
type இனக்குழுவில் மரபுவழிப் பெற்ற உறுப்புகள்			
type இனக்குழுவில் வரையறுக்கப்பட்டவை			

அட்டவணை 4 type **இனக்குழு**

statement இனக் குழுவின் உறுப்புகள்	**உறுப்புகளின் அணுகியல்பு**		
	private	protected	public
statement இனக்குழுவில் மரபுவழிப் பெற்ற உறுப்புகள்			
statement இனக்குழுவில் வரையறுக்கப்பட்டவை			

அட்டவணை 5 statement **இனக்குழு**

பொருள்	**அணுகமுடியும் உறுப்புகள்**	
	தரவு உறுப்புகள்	**உறுப்புச் செயற்கூறுகள்**
type இனக்குழு		
statement இனக்குழு		

அட்டவணை –6 பொருள்

2) கீழ்க்காணும் நிரலில் உள்ள பிழைகளைச் சுட்டி, காரணம் கூறுக.

```
#include<iostream.h>
class A
{
        private :
                int a1;
        public:
                int a2;
        protected:
                int a3;
};
class B : public A
{
```

```
        public:
        void func()
    {
            int b1, b2 , b3;
            b1 = a1;
            b2 = a2;
            b3 = a3;
    }
    };
    void main()
    {
            B der;
            der.a3 = 0;
            der.func();
    }
```

3) கீழ்க்காணும் அறிவிப்புகளை நோக்குக, கீழுள்ள வினாக்களுக்கு விடை தருக.

```
    class vehicle
  {
        int wheels;
        public:
            void inputdata( int, int);
            void outputdata();
        protected :
            int passenger;
  };
  class heavy _vehicle : protected vehicle
  {
            int diesel_petrol;
            protected:
            int load;
        public:
            void readdata( int, int);
            void writedata();
  };
  class bus: private heavy_vehicle
```

```
{
        char marks[20];
        public:
                void fetchdata( char );
                void displaydata();
};
```

அ) heavy vehicle என்னும் இனக்குழுவின் அடிப்படை இனக்குழுவையும் தருவிக்கப்பட்ட இனக்குழுவையும் குறிப்பிடுக.

ஆ) displaydata() என்னும் செயற்கூறு மூலம் அணுக முடிகிற தரவு உறுப்புகளைக் குறிப்பிடுக.

இ) bus இனக்குழுவின் பொருள், அணுக முடிகிற தரவு உறுப்புகளைக் குறிப்பிடுக.

ஈ) outputdata() என்னும் செயற்கூறினை, heavy_vehicle இனக்குழுவின் பொருள்களால் அணுக முடியுமா **?**

4) கீழ்க்காணும் நிரலின் வெளியீடு என்னவாக இருக்கும்.

```
#include<iostream.h>
#include<conio.h>
class student
{
                int m1, m2, total;
        public:
                student ( int a, int b)
{
                m1 = a;
                m2 = b;
                cout<<"\n parameterized constructors..";
  }
};
```

அதிகாரம் 19

சமுதாயத்தின் மீது கணிப்பொறியின் தாக்கம்

(Impact of Computers on Society)

19.1 முன்னுரை

“இந்தியா அதன் ஏழு லட்சம் கிராமங்களில் வாழ்கிறது”

- மோகன் தாஸ் கரம்சந்த் காந்தி

தகவல் தொழில்நுட்பத்தின் பலன்கள் சாதாரண மனிதர்களையும் சென்றடைய வேண்டுமெனில் குறைந்தபட்சம் மூன்று தொழில்நுட்பக் கூறுகள் நமக்குத் தேவை :

- இணைப்பாக்கம் (கணிபொறிப் பிணையங்கள் மற்றும் இணைய வசதி)
- விலை மலிவான கணிப்பொறிகள் அல்லது அதுபோன்ற சாதனங்கள்
- மென்பொருள்

நம் கவனத்தை ஈர்க்கும் தகவல் என்னவெனில், இன்றைய கணிப்பொறிப் பயன்பாடுகளில் 85% **சொல் செயலாக்கம்** (Word Processing) என்பதே, ஆனால், கணிப்பொறியானது சாதாரண மனிதர்க்கு இதைவிட அதிகமாகப் பயன்பட முடியும். தரம் மிக்க தகவல் தொழில்நுட்பக் கல்வி, சாதாரண மனிதர்களும் கணிப்பொறியைச் சிறந்த முறையில் பயன்படுத்த உதவும். சமுதாயத்தை மேம்படுத்தக் கணிப்பொறிகளைப் பயன்படுத்தும் வழிமுறைகளை இந்தப் பாடத்தில் காண்போம்.

19.2 சொந்தப் பயன்பாட்டுக் கணிப்பொறிகள்

சொந்தக் கணிப்பொறிகள் (Personal Computers) நமது வேலை முறையையும், வாழ்க்கை முறையையும், ஏன் சிந்தனையையும் கூட முற்றிலும் மாற்றிவிட்டது. **சொல் செயலாக்கம் (Word Processing)**, **தரவுத் தளங்கள் (Databases)**, **விரிதாள்கள்(spread**

sheets), பல்லூடக முன்வைப்புத் தொகுப்புகள் (Multimedia presentation packages) ஆகியவை பணிகளில் நமது செயல்திறனை அதிகரித்துள்ளன. இன்னும் எத்தனையோ மென்பொருள் தொகுப்புகள் பயன்பாட்டில் உள்ளன. கணிப்பொறி பதிப்பாக்கம் (Desktop publishing) மற்றும் வரைகலைக்கான மனங்கவர் தொகுப்புகள் நாம் செய்யும் பணியின் மதிப்பைக் கூட்டியுள்ளன. ஓவியம் (paint), விளையாட்டு (games)மற்றும் இவை போன்ற ஏராளமான மென்பொருள் தொகுப்புகள் சிறியவர் முதல் முதியவர் வரை அனைத்து வயதினரும் கணிப்பொறியைப் பயன்படுத்தவதற்கான வசதியை நல்குகின்றன. இணைய உலா (Browsing), மின்னஞ்சல் (e-mail), அரட்டை(chat) ஆகியவை நமது வாழ்க்கை முறையை மாற்றிவிட்டன.

இன்றைக்குக் கணிப்பொறிகள் பல்வேறு அளவுகளில், வடிவங்களில் கிடைக்கின்றன. அடிப்படைக் கணிப்பொறியை பயனரின் வீடுகளில் மிக்க பயனுள்ளதாக்கும் வகையில் தகவமைத்துக்கொள்ள முடியும்.

19.3 கணிப்பொறி மயமாக்கப்பட்ட வீடுகள்

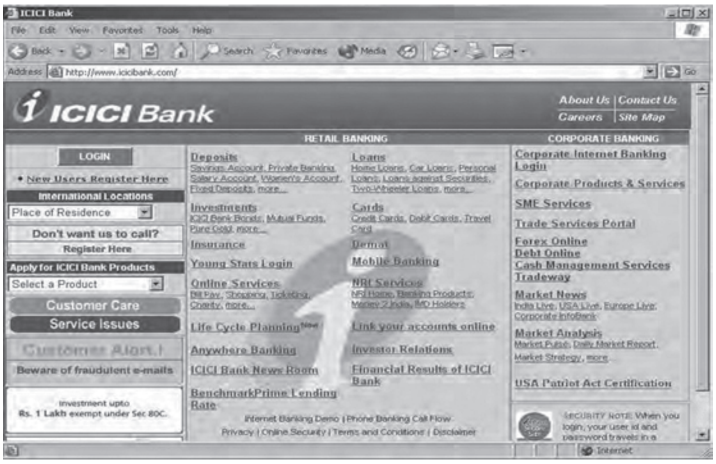
வீடு	தயாரிப்புப் பொருள்களும் சுருக்கமான விளக்கமும்
வரவேற்பறை	✸ எல்சிடி திரை, ஒளிப்படக்கருவி (கேமரா), ஒலி பெருக்கிகள்: சுவரில் பொருத்தப்பட்டிருக்கும். வெளிப்பாடு சிறப்பாய் இருக்கும். தரையிடம் மிச்சமாகும். ✸ தரவுக் காப்பகம் : தரவுகளைச் சேமித்து வைக்கவும் மேலாண்மை செய்யவும் உதவும். ✸ உணர்ச்சிக் கொள்கலன்கள் : சிறிய பெட்டிகள்;ஒரு திரையையும் ஒலிபெருக்கியையும் கொண்டிருக்கும். உணர்வுக்கு இதமூட்டும் நறுமணம் வீசும். மக்களைத் தீய பழக்க வழக்கத்துக்கு ஆட்படாமல் தடுக்கும். ✸ சொந்தத் தரவுக் காப்பகம்: குடும்ப ஒளிப்படங்கள் போன்ற சொந்தத் தகவல்கள் மற்றும் பிற சொந்த அரும்பொருட் செல்வங்கள் போன்றவற்றைச் சேமித்து வைக்கலாம். அதோடுகூட பிற மனிதர்களோடு இணைப்பையும் கொண்டிருக்கும். ✸ படப்பேசியும் கையேடும்: படங்களோடு கூடிய தொலைபேசி எண்களடங்கிய திரட்டு.
குழந்தைகள் அறை	✸ வீட்டிலுள்ள வானொலி, தொலைக்காட்சிப் பெட்டியிலிருந்து வரும் பாடல்களைக் கேட்கும் வசதி கொண்ட சாதனங்கள். ✸ பின்னணி இசையோடு இணைந்து பாடக்கூடிய ‘கரோக்கி’ வசதியுள்ள சாதனங்கள். ✸ மின்–வளர்ப்பு விலங்குகளாகச் செயல்படும் ரோபோக்கள் ✸ இணையத்தில் விளையாட்டுகளை ஆட உதவும் சாதனங்கள்.... இவை தவிர நடப்பு வாழ்வின் பாத்திரங்களைக் கணிப்பொறி உலகின் பாத்திரங்களாக மாற்றி அவற்றோடு குழந்தைகள் விளையாட முடியும்.

வீட்டு அலுவலகம்	✹ அசைவூட்டக் கதைகளை உருவாக்குவதற்கான மென்பொருள்கள் ✹ நினைவுச் சட்டகம்– தொடுதிரை, வருடி, நுண்பேசி வழியாகப் பிற மக்களோட எளிதாக ஊடாட உதவும். ✹ புத்தக அலமாரி : மின்னணு நூல்களைப் படிக்கவும், பாதுகாக்கவும பயன்படும். ✹ சொந்தப் படைப்பாக்கக் கருவி : படம், ஒலி, அசைவூட்டம், நிகழ்படம் போன்ற பல்லூடகத் தகவலுடன் பணியாற்றுதல். உயர்நிலை தரவு அணுகல் முறைமைகள்.
படுக்கை அறை	✹ பல்வேறு சாதனங்களுக்கான தொடு/குரல் கட்டுப்பாடுகள் ✹ காட்சித் திரைகள், தனிச் சிறப்பான தலைபேசிகள், நகர்பேசிகள் ✹ படம்பெருக்கி டிவி– தொலைக்காட்சிப் படங்களை மேற்கூரை அல்லது சுவர்களில் பெரிதாக்கிக் காட்டும். ✹ விழிப்பூட்டு கடிகாரம்
குளியலறை	✹ ஆடிகள், மருந்துப் பெட்டி, தனி சிறப்பான ஒலிபெருக்கிகள்
சமையலறை	✹ ஒலிபெருக்கிகள், அலமாரித் தொலைபேசி, நுண்ணறிவுள்ள காப்புத்துணி, கெட்டில், டோஸ்டர், உணவுப் பகுப்பாய்வி, உடல்நலக் கண்காணிப்பி, உணவுப் பாதுகாப்பு சாதனங்கள்
உணவுக்கூடம்	✹ உணவுப் பொருட்களைச் சூடாக வைத்திருக்க ஊடாடு மேசைத் துணி ✹ பீங்கானில் செய்த கேட்பொலி இயக்கிகளும் ஒலிபெருக்கிகளும் ✹ உணவு மேசையைக் சுற்றிலும் தகவல் தொடர்பு வசதிகள் ✹ சமையல்காரர் மற்றும் சமையலறைப் பணியாளர்களிடம் உரையாட முடிகிற ஊடாடு திரைகள்.

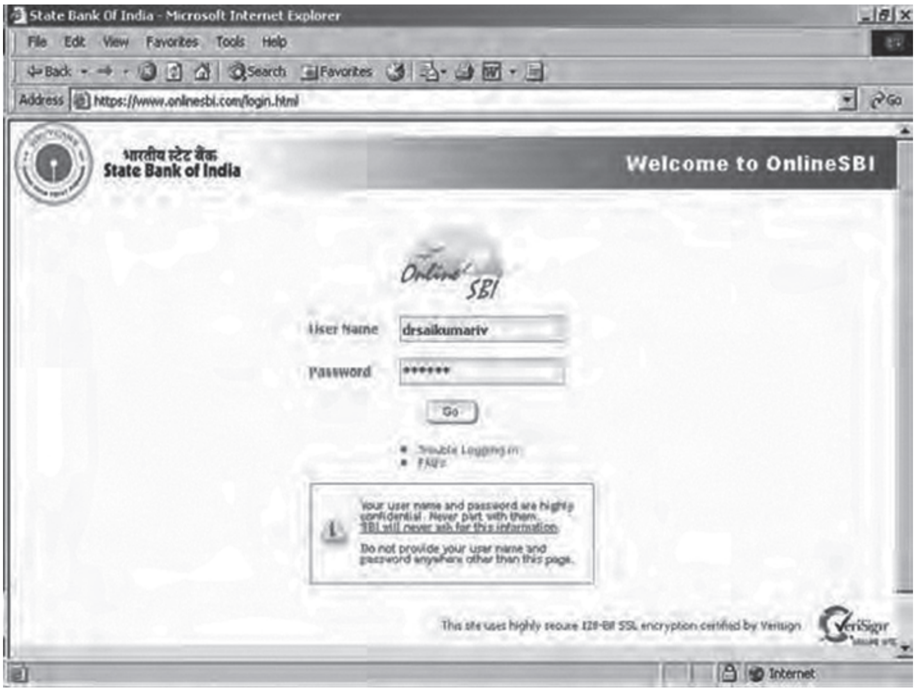
19.4 வீட்டிலிருந்தே வங்கிப் பணியும் கடைப்பொருள் வாங்கலும்

மரபு வழியான வங்கிப் பணியில் பயனாளர் பணப் பரிமாற்றம் போன்ற செயல்பாடுகளை மேற்கொள்ள வங்கிக்கு நேரடியாகச் செல்லவேண்டும். பணப் பரிமாற்றம் என்பது பணத்தைச் சேமிப்பில் போடுவது அல்லது சேமிப்பிலிருந்து பணத்தை எடுத்தல் அல்லது கடன் பெறுதல் போன்ற நடவடிக்கையாக இருக்கலாம். இப்போதெல்லாம் வங்கிகள் வேறுபல சேவைகளையும் வழங்குகின்றன. குறித்த கால வைப்புநிதிகள், வேளாண்மைக் கடன்கள், பாதுகாப்புப் பெட்டகங்கள் மற்றும் தொலைபேசி, மின்சாரம் போன்ற, பிற சேவைகளுக்குக் கட்டணம் செலுத்துதல் ஆகிய வசதிகள் உள்ளன. வங்கிச் செயல்பாடுகளில் சாதாரண மனிதனின் நம்பிக்கை மேம்பட்டுள்ளது. இதன் காரணமாய், தேசியப் பொருளாதாரத்தில் வங்கிகள் முக்கிய கூறாக மாறிவிட்டன.

பணி நேரங்களில் வங்கிகளில் மிக நீண்ட வரிசை நிற்பதைக் காணலாம். வங்கிகளில் தகவல் தொழில்நுட்பம் அறிமுகப்படுத்தப்பட்டதால், பயனாளருக்குச் சேவை வழங்குவதற்குத் தேவையான நேரம் வெகுவாகக் குறைந்துள்ளது. மிக நீண்ட வரிசைகளும் மிக வேகமாகக் கையாளப்படுகின்றன.



படம் 19.1 ஐசிஐசிஐ வங்கி



படம் 19.2 பாரத ஸ்டேட் வங்கி

ஒரு குறிப்பிட்ட வங்கியின் கணக்கிலிருந்து எந்த நேரத்திலும் எந்த இடத்திலும் பணம் எடுத்துக் கொள்ள ஏடீஎம் (ATM - Automatic Teller Machine) போன்ற உயர்தொழில்நுட்ப எந்திரங்கள் பயன்படுகின்றன. பயனாளருக்கு இரவிலோ, விடுமுறை நாட்களிலோ அவசரமாகப் பணம் தேவைப்படும் சூழ்நிலைகளில் இவை உதவுகின்றன. என்றாலும், பயனாளர் அருகிலுள்ள ஏடிஎம் மையத்துக்கு நேரில் செல்ல வேண்டும. மக்கள் பரவலாகப் பயன்படுத்தும் வங்கிகளின் ஒவ்வொரு கிளையும் ஏடீஎம் மையங்களை நிறுவும் நாள் வெகுதொலைவில் இல்லை.

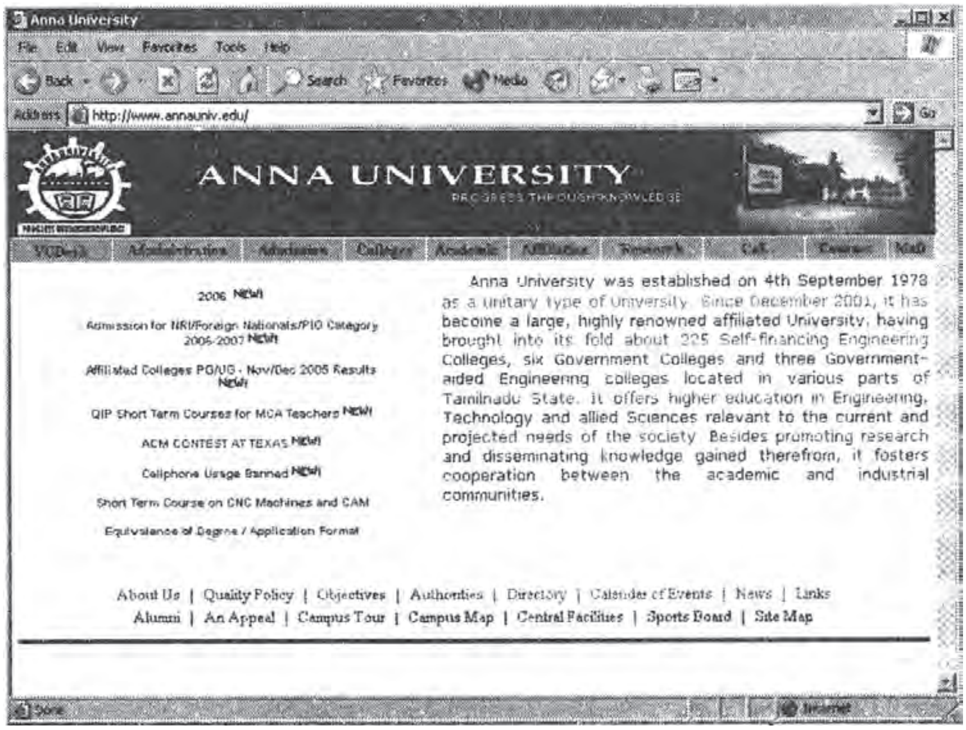
மின்–வங்கிச் சேவை (e-Banking), மக்கள் விட்டிலிருந்தபடியே இணையத்தின் வழியாக வங்கிச் சேவைகளை நுகர வாய்ப்பளித்துள்ளது. இது, வங்கிகளின் வீச்சையும் சேவைகளையும் மெய்யாகவே மேம்படுத்தியுள்ளது எனலாம்.

கணிப்பொறிகள் பல்வேறு பணிகளுக்கும் பயன்படுத்தப்படுகின்றன. கடையில் பொருள்கள் வாங்கவும்கூட பயன்படுத்த முடியும். மின் கடைச் செலவு (e-shopping) முறையில் எந்தப் பொருளையும், எந்த நிறுவனத்தின் பொருளையும், எவ்வளவு வேண்டுமானாலும், எந்த இடத்திலிருந்தும் வாங்கிக் கொள்ள முடியும். நீங்கள் கடைக்கு நேரில் செல்ல வேண்டியதில்லை. கடையின் வலையகத்தில் (website) நீங்கள் வாங்க விரும்பும் பொருட்களின் படம் மற்றும் பிற விவரங்களைக் காணலாம். அவற்றில் தேவையானவற்றைத் தேர்ந்தெடுத்து, அனுப்பி வைக்குமாறு கோரிக்கை அனுப்பலாம். பொருள்களுக்கான விலையை முன்கூட்டியே செலுத்துவதற்குப் பல்வேறு வழிமுறைகள் உள்ளன. கடன் அட்டைகள் (Credit cards), கடைக்காரரிடம் முன்கூட்டியே பதிவு செய்தல் போன்ற வழிமுறைகள் பரவலாகப் பயன்பாட்டில் உள்ளன. வாங்கப்பட்ட பொருட்கள் வீட்டிற்கு வந்து வினியோகிக்கப்படும்.

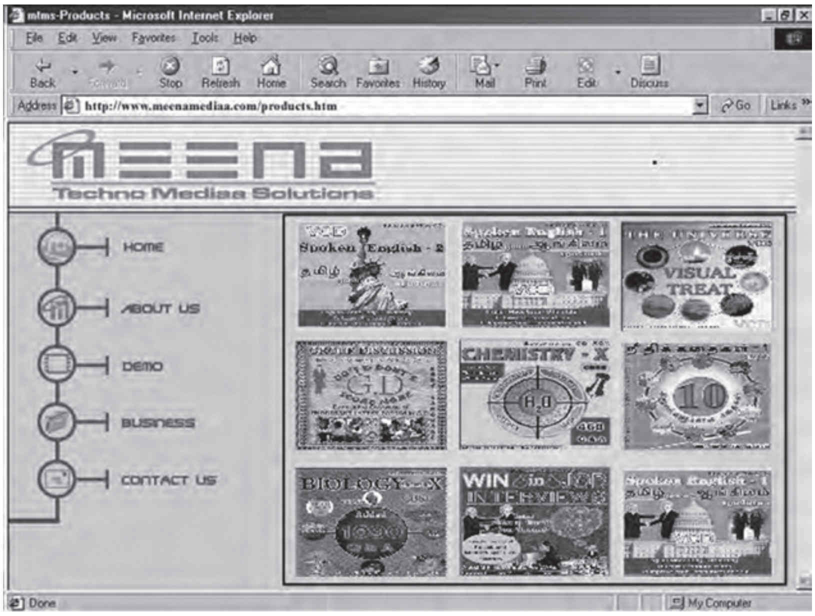
19.5 கல்வியில் கணிப்பொறிகள்

கல்வியில் பல பிரிவுகளிலும் கணிப்பொறிகள் பயன்படுத்தப்படுகின்றன. அவற்றுள் சில.

- உள்நாட்டு/வெளிநாட்டு நூலாசிரியர்களின் கல்விக் குறுவட்டுகளையும், புத்தகங்களின் புதிய பதிப்புகளையும் இணையத்தின் வழியே தேடி அறிந்து அவற்றை வாங்குதல்.
- கணிப்பொறி வரியான பாடப் பயிற்சிகள்(Computer Based Tutorials -CBT).
- பள்ளிகள், கல்லூரிகள், பல்கலைக் கழகங்கள், அவை வழங்கும் படிப்புகள், அவற்றில் அசர்வதற்கான விதிமுறைகள், விடுதி வசதிகள், கல்வி உதவித் தொகைகள், கல்விக் கடன்கள், வேலைவாய்ப்புக்கான வழிகாட்டுதல்– இவை பற்றிய தகவல்களைப் பரப்புதல்.
- மின் கற்றல் (e-Learning) முறையில் இணையத்தின் வழியே கல்வி கற்றுப் பட்டங்களும் சான்றிதழ்களும் பெற முடியும்.



படம் 19.3 அண்ணா பல்கலைக் கழகம்



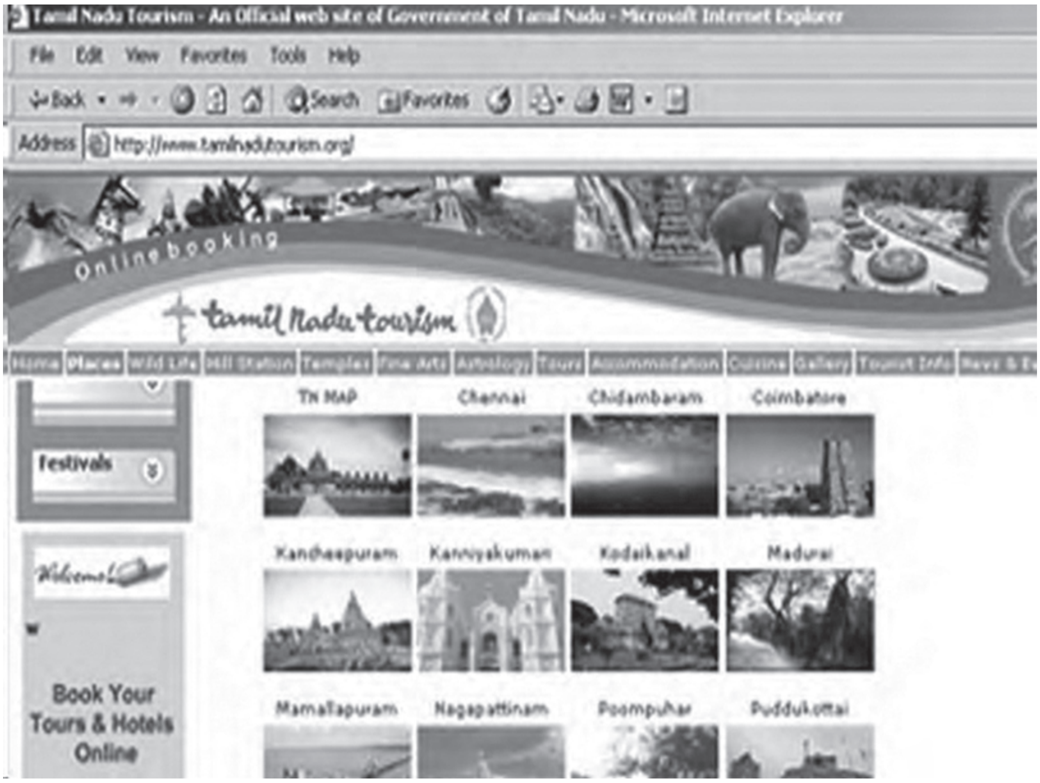
படம் 19.4 கணிப்பொறிவழி பாடப் பயிற்சி

படம் 19.5 உலகளாவிய கல்வி பற்றிய தகவல்

19.6 பொழுது போக்கில் கணிப்பொறிகள்

பொழுதுபோக்கிலும் கணிப்பொறிகளின் பங்களிப்பு உள்ளது. கணிப்பொறி மற்றும் இணைய மூலமாக ஓவியம், இசை, நடனம், யோகா, விளையாட்டுகள், அறிவியல், இயற்கை, சூடான செய்திகள் மற்றும் நிகழ்வுகள் பற்றிய உங்கள் அறிவைப் புதுப்பித்துக் கொள்ள முடியும். பல்வேறு வழிபாட்டுத் தலங்கள் பற்றியும், சுற்றுலாத் தலங்கள் பற்றியும் நிறையவே அறிந்து கொள்ள முடியும்.

படம் 19.6 பொழுதுபோக்கில் கணிப்பொறிகள்



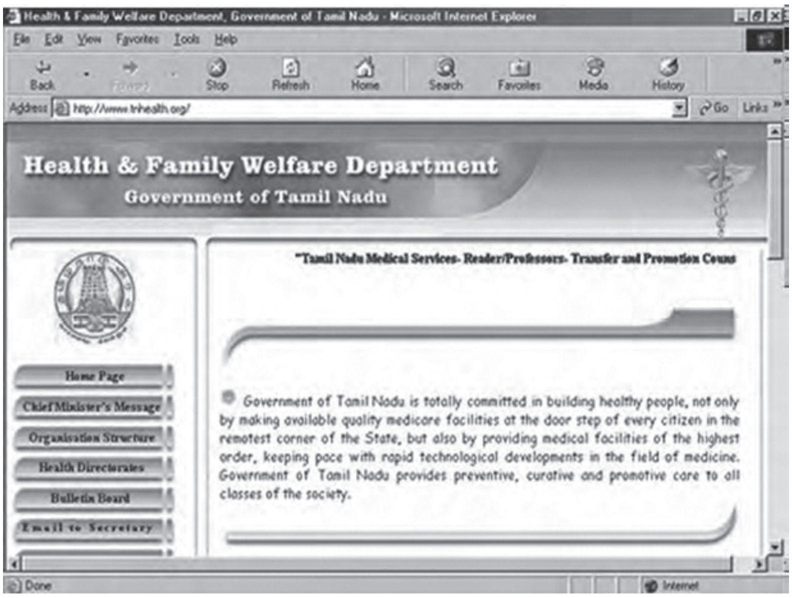
படம் 19.7 சுற்றுலாத் துறையில் கணிப்பொறிகள் (தமிழ்நாடு)

படம் 19.8 சுற்றுலாத் துறையில் கணிப்பொறிகள் (இந்தியா)

19.7 நலவாழ்வுக்குக் கணிப்பொறிகள்

நலவாழ்வுத் துறையில் ஏராளமான தரவுகள் குவிந்து கிடக்கின்றன. அவற்றைப் பராமாரிக்கக் குறைந்த அளவு நிதி வளமும் மனித வளமுமே உள்ளன. அவற்றைக் கட்டிக் காப்பதற்கான தேவை அதிகரித்துள்ளது.

மருத்துவர்களும் மருத்துவத் துறை நிர்வாகிகளும் கணிப்பொறிகளைப் பயன்படுத்தத் தொடங்கிய பிறகு நலவாழ்வுச் சேவைகள் மேம்பட்டுள்ளன.



படம் 19. 9 நலவாழ்வு மற்றும் குடும்பநலத் துறையில் கணிப்பொறிகள் (இந்தியா)

நலவாழ்வுத் துறையின் பல பிரிவுகளிலும் கணிப்பொறிகள் பயன்படுத்தப்படுகின்றன. அவற்றுள் சில–

- மருத்தவமனை மேலாண்மை முறைமை
- நோயாளிகள் தொடர்சேவை முறைமை
- நோயாய்வு ஆவணங்களை நலவாழ்வு நிலையங்களுக்கிடையே பரிமாறிக் கொள்ளல்.
- தொற்று நோய் பற்றி முழுவிவரம் அறிதலும் கண்காணித்தலும்
- உயர்நிலைக் கணிப்பணி நுட்பங்களுடன் கூடிய தீர்மானிப்பு உதவி முறைமை

இன்றைக்குப் பெரும்பாலான மருத்துவர்கள் தமது தேவைகளுக்கு ஏற்ற வகையில் நவீன வழிமுறைகளைப் பின்பற்றுகின்றனர். நோயாளிகளக்கு இது ஒரு நல்ல அறிகுறியாகும். தொலைமருத்துவம் (Tele-medicine) என்பது, பெரும்பாலும் மேலே குறிப்பிடப்பட்டுள்ள அடிப்படை முறைமைகளின் மீதே கட்டமைக்கப்பட்டுள்ளது. சேய்மையிலிருந்து கொண்டே நோயாய்வு செய்ய இணையம் வழி வகுக்கிறது. வல்லுநர் ஆலோசனை கிடைக்க முடியாத இடங்களிலும் அத்தகைய ஆலோசனை கிடைப்பதை இணையம் உறுதி செய்கிறது.

19.8 வேளாண்மைத் துறையில் கணிப்பொறிகள்

வேளாண்மைத் துறை, தொழில்நுட்பம் சாராத துறைபோலத் தோன்றலாம். சாதாரணமாக இத்துறையை நோக்குபவர்கள் எண்ணுவதை விடவும் கணிப்பொறி யமாக்கத்தால் இத்துறை மிகவும் பலனடைந்துள்ளது. செய்தொழிலாகவும், விருப்பார்வமாகவும் வேளாண்மையில் ஈடுபட்டுள்ள உழவர்கள், விதைகள் பற்றிய கணிப்புகள் மற்றும் பூச்சிக் கொல்லிகள் பற்றிய தகவல்கள் அடங்கிய நிகழ்நிலைத் தகவல் வளங்களான இணைய தளங்கள் மூலம் பெரிதும் பயனடைந்துள்ளனர். தட்பவெப்பநிலை, மண்வளம், தற்போதைய சந்தை விலைகள் ஆகியவற்றின் அடிப்படையில் எந்த விளைபொருள் அதிக அளவு லாபத்தைத் தரும் என்பதைக் கணித்துச் சொல்லும் வருமானக் கணிப்புத் தகவல்கள், உழவர்களுக்குப் பெரிதும் பயன்படுகின்றன.

படம் 19. 10 வேளாண்மையில் கணிப்பொறிகள்

வேளாண்மைத் துறையில் மென்பொருள் உருவாக்கப்பட்டுள்ள பிரிவுகள் :

- வேளாண்மை நிதியங்கள் மற்றும் கணக்கியல்
- மாற்று விவசாய நுட்பங்கள்
- கால்நடை வளர்ச்சி
- கட்டிடங்கள் மற்றும் நீர்ப்பாசனம்
- பிற வேளாண்மைத் துறையினர் மற்றும் விஞ்ஞானிகளுடன் கலந்துரையாடல்
- பண்ணை நில மதிப்பீடுகள்
- உரப் பகுப்பாய்வு
- இணையத்தில் பண்ணை வளங்கள், அரட்டைக் குழுக்கள், வகையின விளம்பரங்கள், பிற பண்ணை தொடர்பான தகவல்கள் – இவற்றுக்கான தொடுப்புகளைக் கண்டறிதல்.
- தோட்டக் கலை
- பசு இனத்தை மேம்படுத்தி வருமானத்தைப் பெருக்குதல்
- நில மேலாண்மை
- பால் உற்பத்தி
- செயற்கைக்கோள் படங்களைப் பயன்படுத்தி விளைபயிர் பற்றிய முடிவுகளை மேற்கொள்ளுதல்

19.9 நிகழ்நேரப் பயன்பாடுகளில் இணையம்

மேற்சொல்லப்பட்ட அனைத்துப் பயன்பாடுகளும் நிகழ்நேரத்தில் இணையத்தில் கிடைப்பவை. இரயில், விமானப் பயணச் சீட்டுகளை வீட்டில் அமர்ந்தபடியே நிதானமாக இணையம் மூலம் முன்பதிவு செய்து கொள்ள முடியும்.

படம் 19.11 நிகழ்நேரப் பயன்பாடுகள்

பயிற்சிகள்

இந்தப் பாடத்தில் விளக்கப்பட்ட பயன்பாட்டு மென்பொருள்கள் பற்றி மேலும் புரிந்து கொள்ள உதவும், பல்லூடக விளக்கப் படங்கள் தனியாக ஒரு குறுவட்டில் உங்கள் பள்ளிக்குத் தரப்பட்டுள்ளது. அதன் உள்ளடக்கத்தை நீங்கள் கட்டாயம் பார்க்க வேண்டும். இக்குறுவட்டைப் பெற உங்கள் ஆசிரியரை அணுகுக. பாடத்தில் குறிப்பிடப்பட்டுள்ள இணைய தளங்களையும் பார்வையிடுங்கள்

அதிகாரம் 20

தகவல் தொழில்நுட்பம் சார்ந்த சேவைகள்

(IT Enabled Services)

20.1 முன்னுரை

பல்வேறு துறைகளில் சேவைகளின் தரத்தை மேம்படுத்திக் கொள்ளப் பயனாளர்களுக்குத் தகவல் தொழில்நுட்பம் உதவுகிறது. இத்தகைய சேவைகளைத் **தகவல் தொழில்நுட்பம் சார்ந்த சேவைகள்** (IT Enabled Services - ITES) என்கிறோம். இத்தகு சேவைகள் பெரும்பாலும் அதிகமான மனிதர்களை ஈடுபடுத்துகின்ற சேவைகளாகும். தொலைத் தொடர்புப் பிணையங்கள் அல்லது இணையம் வழியாக இச்சேவைகள் பலதரப்பட்ட வணிகப் பிரிவுகளுக்கும் வழங்கப்படுகின்றன. தகவல் தொழில்நுட்பம் சார்ந்த சேவைகள் பெருமளவு வேலைவாய்ப்புகளை அதிகரித்துள்ளன.

கணிப்பொறியைப் பயன்படுத்தி ஒரு கடிதத்தைத் தட்டச்சு செய்வது ஐடீஇஎஸ் சேவையில் அடங்குமா? இல்லை என்பதே பதிலாகும். எனினும் பயனாளர் **டிக்டாஃபோன்** (Dictaphone)என்னும் சிறப்புச் சாதனத்தில் பேசி, அப்பேச்சினை அப்படியே உரைவடிவில் கடிதமாக மாற்றும் பணி ஐடீஇஎஸ் இல் அடங்கும்.

சொல்செயலிகள் (Word processors), விரிதாள்கள்**(spread sheets)**, தரவுத் தளங்கள், (databases) ஆகியவை, பல்வேறு மரவுவழிப்பட்ட சேவைகள் தகவல் தொழில் நுட்பம் சார்ந்தவையாக மாற வழி வகுத்துள்ளன. எனினும், பயனாளர், இத்தகைய தகவல் தொழில்நுட்பக் கருவிகளைப் பயன்படுத்திப் பலன்களைப் பெறவேண்டுமாயின் அவற்றின் பல்வேறு கூறுகளையும் கற்றுக் கொள்ள வேண்டும். ஐடீஇஎஸ், பயனாளர்கள் இவற்றைக் கற்க வேண்டிய தேவையை குறைக்கின்றது. அதன்மூலம் அவற்றின் மதிப்பைக் கூட்டுகின்றது. ஆக, தகவல் தொழில்நுட்பத்தை அதிகம் கற்றறியாத பயனாளர்களும் தகவல் தொழில் நுட்பத்தின் சக்தி முழுவதையும் கைவரப்பெற ஐடீஇஎஸ் உதவும் வல்லமையைப் பெற்றுள்ளது.

ஐடீஇஎஸ் சேவையின் தரத்தை நேரடியாகவோ, மறைமுகமாகவோ மேம்படுத்துகிறது. வாடிக்கையாளர்களின், கூடுதலான திருப்தி, சிறந்த தோற்றம் உணர்வும், மேம்பட்ட தரவுத்தளம் ஆகியவை நேரடியான பலன்களில் சிலவாகும். மறைமுகப் பலன்கள் சிறிதுகாலம் கழித்துக் கிடைக்கப்பெறும். ஒரு பயனுக்கென சேகரிக்கப்பட்ட தரவுகள் வேறுபல தேவைகளுக்கும் சிறிது காலத்துக்குக் பின் பயன்படும்.

இந்தப் பாடத்தில் தரப்பட்டுள்ள தகவல் தொழில்நுட்பம் சார்ந்த சேவைகளுள் சில–

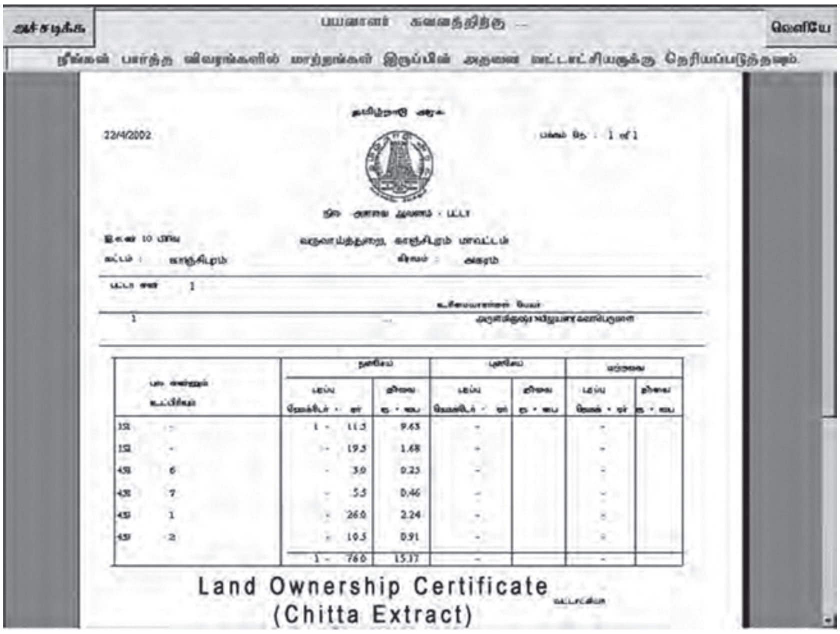
- மின் அரசாண்மை (e-Governance)
- அழைப்புதவி மையங்கள் (Call Centres)
- தரவு மேலாண்மை (Data Management)
- தொலை மருத்துவம் (Telemedicine)
- மருத்துவ ஆவணப் பெயர்ப்பு (Medical Traanscription)
- கணிப்பொறித் தரவாக்கம் (Data Digitilization)
- வலையகச் சேவைகள் (Website Services)

வணிகச் செயலாக்கப் புறத்திறனீட்டம் (Bussiness Process Outsourcing-BPO), துடிமத் தகவல் உருவாக்கம் (Digital Content Development), அசைவூட்டம்(Animation), மனிதவளச் சேவைகள் (Human Resource Services), சேய்மைப் பராமரிப்பு (Remote Maintenance) ஆகியவை ஐடீஇஎஸ் ஊடுருவியுள்ள பிற சேவைகளாகும்.

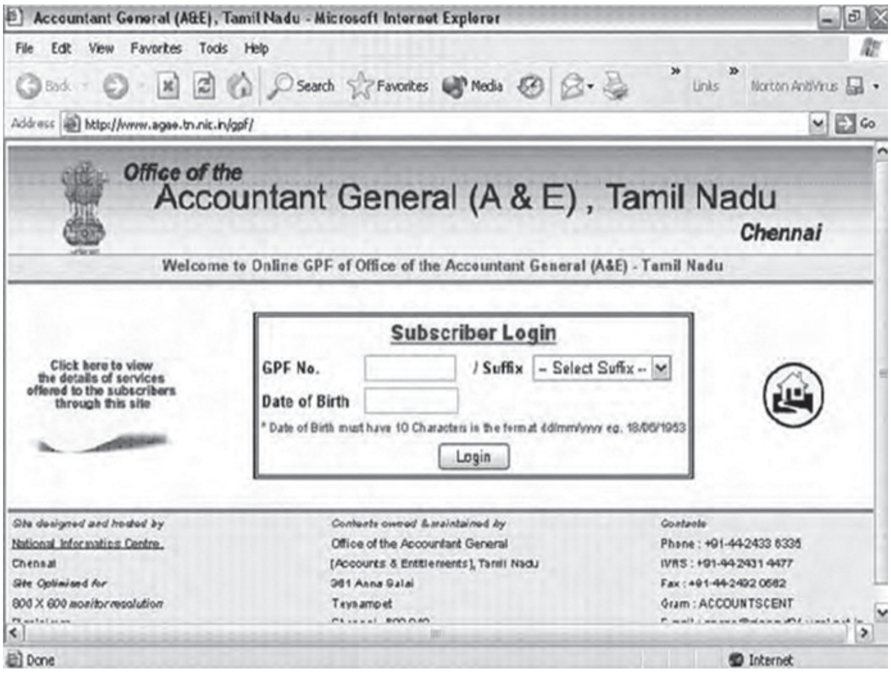
ஐடீஇஎஸ் பணிகளுக்கு, தகவல் தொழில்நுட்பத்தில் குறிப்பாக தரவுத் தளங்களிலும், இணையத்திலும் செய்முறை அறிவுத்திறன் தேவையாகும். தவிரவும், ஆங்கிலத்தில் சிறந்த தகவல் தொடர்ப்புத்திறன் (Communication skill) இருக்க வேண்டிய கட்டாயத் தேவையாகும். ஐடீஇஎஸ் சேவைகளை திறம்பட நடைமுறைப்படுத்துவதற்கு, தொழிலகப் பயன்பாடு, செய்தொழில் நுட்பம் (Professionalism), நற்பழக்கம் (etiqutte) ஆகியவற்றின் அடிப்படைகளை நன்கு புரிந்துக் கொள்ளக்கூடிய அகத்திறனில்(Softskill) முறையான பயிற்சி அளிக்கப்பட வேண்டும்.

20.2 மின் அரசாண்மை (e-Governance)

நீங்கள் அரசு வலையகங்களைப் பார்வையிடவும் அவை வழங்கும் சேவைகளைப் பெறவும் கணிப்பொறிகள் உங்களுக்கு உதவுகின்றன. அரசு அமைத்துள்ள பல்வேறு வலையகங்கள் பல பயனுள்ள தகவல்களைக் கொண்டுள்ளன. அரசுத் துறைகள் பற்றி விவரங்கள், குறிப்பிட்ட பணிகள், சிறப்பு நலத்திட்டங்கள், ஆவணங்கள், தொடர்புகள், தொடுப்புகள், ஐஏஎஸ் அக இணையம், வலையக வரைப்படம், தேடல், புதியது என்ன, பத்திரிகைக் குறிப்புகள், கருத்தாய்வு ஆகியவற்றைக் கொண்டுள்ளன. இந்த வலையகங்கள் ஆங்கிலம், தமிழ் ஆகிய இரு மொழிகளிலும் உள்ளன.



படம் 20.1 இணையத்தின் மூலமாக நிலச் சான்றிதழ் பெறுதல்



படம் 20. 2 தலைமைக் கணக்காயர் (ஏ & இ) அலுவலகம், தமிழ் நாடு அரசு.

20.3 அழைப்புதவி மையங்கள் (Call Centres)

தகவல் தொழில்நுட்பத்தின் தாக்கம் உலகம் முழுவதும் காணப்படுகிறது. தகவல் தொழில்நுட்பச் சேவைகளின் பயனாளர்கள் அல்லது வாடிக்கையாளர்கள் உலகம் முழுவதிலும் நிறைந்துள்ளனர். ஆண்டில் 365 நாட்களும், நாளில் 24 மணிநேரமும் தகவல் தொடர்பு சேவையை நிறுவனங்கள் வழங்க வேண்டும் என வாடிக்கையாளர்கள் எதிர்பார்கின்றனர்.

அழைப்புதவி மையம் என்பது, சில வேளைகளில், குறிப்பிட்ட வாடிக்கையாளர் சேவைக்காக தொலைபேசி அடிப்படையில் அமைந்த சேவைப் பகிர்வு மையம் என வரையறுக்கப்படுகிறது. காரணம் இம்மையங்கள், சந்தைப்படுத்துதல் (Marketing), விற்பனை (selling), தகவல் பரிமாற்றம், வாடிக்கையாளர் தொடர்பான பணிகளுக்குப் பயன்படுத்திக் கொள்ளப்படுகின்றன. ஒவ்வோர் அழைப்புதவி மையமும் போதுமான அளவு தொலைதொடர்பு வசதிகளையும், பயிற்சி பெற்ற ஆலோசகர்களையும் கொண்டுள்ளன. மிகப்பரந்த தரவுத் தளங்கள், இணையம் மற்றும் பிற நிகழ்நேரத் தகவல் உதவி மையங்களையும், அணுகும் வசதிகளைப் பெற்றுள்ளன. வாடிக்கையாளர்களுக்குத் தகவலையும், உதவிச் சேவைகளையும் அழைப்பதவி மையங்கள் வழங்குகின்றன. இவை ஆண்டில் அனைத்து நாட்களிலும், நாளில் அனைத்து நேரங்களிலும் (அதாவது 24x365 சேவை) செயல்படுகின்றன.

20.4 தரவு மேலாண்மை (Data Management)

தரவு மேலாண்மை என்பது, ஐடீஇஎஸ் சேவையில் ஒரு வகை, பல்வேறு மூலங்களிலிருந்து பெறப்படும். தரவுகளைத் திரட்டுவது அவற்றைக் கணிப்பொறியில் சேமிப்பது, பின் செயலாக்குவது ஆகிய பணிகளை உள்ளடக்கியதாகும். மரபுவழி, தரவுச் செயலாக்கச் சேவை என்பது, கையெழுத்தில் நிரப்பப்பட்ட படிவங்களில் உள்ள தரவுகளை கணிப்பொறியில் பதிவது, படங்களையும், அச்சிட்ட வெளியீடுகளையும் கணிப்பொறியில் ஏற்றி அவை அனைத்தையும் ஒன்றாக்கித் தரவுத் தளங்களை உருவாக்குவது ஆகிய பணிகளை உள்ளடக்கியதாகும். ஆனால், பல்லூடகத் தொழில் நுட்பம் மற்றும் இணையத்தின்

வருகையால், தரவுத் தளங்களில் அச்சிட்ட ஆவணங்களை மட்டுமின்றி படங்கள், ஒலி மற்றும் நிகழ்படதகவல்களையும் சேமிக்க முடியும் அதே போல, தகவல் வெளியீட்டு ஊடகங்களிலும் பெருமளவு மாற்றங்கள் ஏற்பட்டுள்ளன. வழங்கிக் (server) கணிப்பொறிகளில் சேமித்து வைக்கப்பட்டுள்ள தரவுத் தளங்களைப் பலரும் பகிர்ந்து கொள்ள முடிகிறது. அச்சிட்ட பதிப்புகளிலும் மற்றும் குறுவட்டுகளிலும் தகவல்கள் வெளியிடப்படுகின்றன. இணையம் வழி பெறப்படும் வினவல்களின் அடிப்படையில் இந்த வெளியீடுகள் அமைகின்றன.

நிறுவனங்களில் செயல்திறனுடன் லாபம் தரத்தக்க வகையில் தகவல் தொழில்நுட்பத்தைப் பயன்படுத்திக் கொள்வதில் தரவு மேலாண்மை முக்கிய இடம் வகிக்கிறது. இந்த வகைப்பாட்டில் வழங்கப்படும் ஐடீஇஎஸ் சேவைகள் வருமாறு :

- ஆஸ்க்கி (ASCII) உரைவடிவில் தகவல்களைத் தரவுத்தளத்தில் பதிவேற்றல்.
- அச்சிட்ட உரை ஆவணங்களைக் கணிப்பொறி ஆவணமாக மாற்றிச் செயலாக்குதல்
- தேவைகேற்ற வகையில் அறிக்கை தயாரித்தல்
- தரவுகள் உள்ளீடு
- தரவுகளைத் திருத்தியமைத்தல்
- ஆவண உருவாக்கம்
- படிவங்களை படிமங்களாகக் குறுவட்டுகளில் பதிதல்
- கையெழுத்து, அச்செழுத்து, காந்த எழுத்து, பட்டைக்குறிமுறை இதுபோன்ற எந்த வடிவத்தில் இருக்கும் தகவல்களையும் கணிப்பொறித் தகவலாக மாற்றிச் சேமித்தல்
- படங்களை வருடிச் சேமித்தல்
- படங்களை உள்ளீடு செய்தல்
- படங்களை கணிப்பொறியில் சேமித்து வைத்தல்; தேவையான போது தேடி எடுத்தல்
- வெளியீடுகளை ஆய்வு செய்தல்
- புள்ளிவிவரப் பகுப்பாய்வு

தரவு மேலாண்மை அடிப்படையிலான ஐடீஇஎஸ் சேவைகள் மூலமாகப் பெரிதும் பயனடையக்கூடிய நிறுவனங்கள் :

- கணக்குவைப்பு, நிதியியல் சேவைகள் போன்ற பின்னணி
  அலுவலகச் செயல்பாடுகள்
- வங்கித் துறை
- அரசுத்துறைகள்
- மருத்துவமனைகள்
- காப்பீட்டுத்துறை
- சட்டத் துறை
- உற்பத்தி நிறுவனங்கள்
- நகராட்சிகள்

- காவல்துறை
- பொதுமக்கள் நுகர்சேவைகள்
- பதிப்புத்துறை
- தொலைதொடர்பு
- போக்குவரத்து

மேலே குறிப்பிட்ட ஒவ்வொரு நிறுவனமும் ஐடீஇஎஸ் சேவைகளில் தரவு மேலாண்மையின் முக்கியமான பிரிவுகளில் பரந்த வாய்ப்புகளை உருவாக்கியுள்ளன. வங்கித்துறை (Banking), நிதியியல் சேவைகள் (Financial services), காப்பீடு (Insurance) ஆகியவை இணைந்து சுருக்கமாக பிஎஃப்எஸ்ஐ (BFSI) என்று அழைக்கப்படுகின்றன. பிஎஃப்எஸ்ஐ மற்றும் ஓய்வூதியச் சேவைகள் ஐடீஇஎஸ்ஸில் பெருமளவில் வளர்ச்சிபெறும் பிரிவுகளாகும்.

இப்பிரிவுகளில், தரவுப் பாதுகாப்பு, வாடிக்கையாளர் தனிமறை (Privacy) ஆகிய இரண்டு முக்கியமான கூறுகளாகும். ஐடீஇஎஸ் சேவையாளர் இவற்றுக்கு உத்திரவாதம் அளிக்க வேண்டும். ஐடீஇஎஸ் சேவையாளர் ஒருவரே பல நிறுவனங்களுங்கும் சேவை வழங்க முடியும். எனவே ஒவ்வொரு நிறுவனத்தின் தனிமறைக் கூறுகளுக்கும் ஐடீஇஎஸ் சேவையாளர் உத்திரவாதம் தர வேண்டும். கணிப்பொறி நன்னெறி ஐடீஇஎஸ்ஸின் வெற்றிக்கு இன்றியமையாததாகும்.

20.5 மருத்துவப் பெயர்ப்பாவணமும் தொலை மருத்துவமும்

(Medical Transcription and Tele-Medicine)

மருத்துவப் பெயர்ப்பாவணம் என்பது, ஒரு மருத்துவ நோய் ஆய்வின் முடிவுகளை விளக்குகின்ற ஒரு நிலையான, சட்டபூர்வமான ஆவணம் ஆகும். இதனை இணையம் வழி உடனுக்குடன் எங்கும் அனுப்பிவைக்க முடியும். காப்பிட்டுத் தொகையைப் பெற இது உதவும். மருத்துவப் பெயர்பாவணம் மூன்று முதன்மையான படிநிலைகளைக் கொண்டது. அவை

படிநிலை –1 : ஐடீஇஎஸ்ஸின் இச்சேவையைப் பயன்படுத்திக் கொள்ள விரும்பும் மருத்துவமனைகள் ஒரு சேவையாளரிம் ஒப்பந்தம் செய்து கொள்கின்றன. மருத்துவர் ஒரு சிறப்புச் சாதனம் மூலம் அல்லது தொலைபேசி வழியாகத் தன்னுடைய ஆலோசனைகளை வழங்குகிறார். குரல் தகவல் மறுமுனையிலுள்ள வழங்கிக் கணிப்பொறியில் (server) பதிவு செய்யப்படுகிறது.

படிநிலை –2 : ஒலித்தகவல் கணிப்பொறித் தகவலாய் மாற்றப்பட்டு ஐடீஇஎஸ் சேவையாளருக்கு அனுப்பப்படுகிறது. பெரும்பாலும் இந்தச் சேவை நிறுவனம் வேறொரு நாட்டில் இருக்கும். அமெரிக்கா போன்ற நாடுகளில் பெயர்ப்பாவணச் சேவையை வழங்குவது, நோயாளிக்கும் மருத்துவமனைக்கும் மிகுந்த செலவுபிடிக்கச் கூடியதாகும். எனவே செலவு கட்டுப்படியாகும் ஒரு நாட்டில் இப்பணியைச் செய்து முடிப்பதன் மூலம், ஐடீஇஎஸ்ஸின் இப்பிரிவு செலவைக் குறைக்கிறது. கணிப்பொறித் தகவல் மீணடும் ஒலித்தகவலாக மாற்றப்படுகிறது. பயிற்சி பெற்ற பெயர்ப்பாவண எழுத்தாளர்கள் ஒலித்தகவலைக் காதில் கேட்டு, கணிப்பொறியில் உரைத்தகவலாய் மாற்றித் தருவர். இந்த ஆவணம் மருத்துவர் செய்த நோயாய்வு பற்றிய முறையான ஆவணமாகப் பாதுகாக்கப்படும்.

படிநிலை –3: இவ்வாறு பெயர்தெழுதப்பட்ட கோப்புகள் தரக்கட்டுப்பாட்டு வல்லுநர்களுக்கு அனுப்பப்படும். மருத்துவரின் குரல் தகவலைக் கேட்டு, பெயர்பாவணத்துடன் சரிபார்ப்பர். தேவையெனில் திருத்தங்கள் செய்வர். அதன்பின் பெயர்த்தெழுதப்பட்ட அறிக்கைகள் உரை ஆவணங்களாக மருத்துவமனைக்கே திரும்பவும் அனுப்பிவைக்கப்படும். இவை சட்டப்படி செல்லுபடியாகும் ஆவணங்களாகும். காப்பீட்டுத் தொகை பெறப் பயன்படுத்திக் கொள்ள முடியும்.

20.6 **கணிப்பொறித் தரவாக்கம்** (Data Digitization)

கணிப்பொறித் தரவாக்கம் என்பது பிற வடிவங்களில் உள்ள தகவல்களைக் கணிப்பொறியில் கையாள்வதற்கேற்ற தகவலாய் மாற்றியமைப்பதாகும். வரைப்படங்கள், கையெழுத்து, நிகழ்ப்படங்கள், ஒலி ஆகிய பல்வேறு வடிவங்களில் உள்ள தகவல்களைக் கணிப்பொறித் தகவலாக்க முடியும்.

கணிப்பொறியில் தகவல்களைச் சேமித்து வைப்பதால் பயனாளர் பல்வேறு பலன்களைப் பெற முடியும். தகவலை எளிதாக அணுக முடியும். தேவையான தகவலை எளிதாகத் தேடிக் கண்டறிந்து எடுத்தாள முடியும். தகவல்களைத் திருத்திப் புதுப்பிக்க முடியும். எனினும், கணிப்பொறியில் தகவலைச் சேமித்துவைப்பதை, நீண்டகாலத் தகவல் காப்பு நுட்பமாகப் பயன்படுத்துவது ஆபத்தில் முடியவும் வாயப்புண்டு. கணிப்பொறித் தகவலாக்கத் தொழில்நுட்பங்கள் வெகுவிரைவாக மாறி வருகின்றன. தகவல் சேமக்காப்பு (preservation) என்பது நீண்டகாலச் செயல்நுட்பம். ஆனால் இன்றைய பல தொழில்நுட்பங்கள் மிகக் குறுகிய காலத்தில் பயனொழிந்து போகின்றன. தொழில்நுட்பத்தின் இத்தகைய நிலையற்ற தன்மை, கணிப்பொறித் தகவல்களின் இழப்பில் கொண்டு போய் விட்டுவிடலாம். தகவலைப் பலகாலம் சேமித்துக் காக்க வேண்டிய நோக்கமே தோற்கடிக்கப்பட்டுவிடும். கணிப்பொறித் தரவாக்கத் தொழில்நுட்பம் பயன்படும் சில பிரிவுகளைக் காண்க:

- ஆண்டறிக்கைளும் விலைப்பட்டியல்களும்
- புத்தகங்கள்
- தரவுத்தள ஆவணக் காப்பகம்
- மின்னஞ்சல் விவரப்பட்டியல்களும் துண்டு வெளியீடுகளும்
- பொறியியல் மற்றும் வடிவாக்கம்
- நிலவியல் தகவல் முறைமை
- திரைப்படங்கள், இசைப்பாடல்கள், மிகுதெளிவு படச் சேமிப்பு
- செய்பொருள்/சேவைக்கான பயிற்சிக் குறிப்பேடுகள்
- ஆய்விதழ்க்ள் மற்றும் கருத்தரங்குக் கட்டுரைகள்

கணிப்பொறித் தரவாக்கத்தின் படிநிலைகள் வருமாறு :

- வாடிக்கையாளரின் தேவைகளை அறிந்து கொள்ளல்
- அவையே கணிப்பொறித் தரவாக்கத்தின் நோக்கங்களை வரையறுப்பதற்கான அடிப்படைகளாகக் கொள்ளப்படுகின்றன.
- ஒரு முன்னோட்டப் பயன்பாடு உருவாக்கப்படுகிறது.

- முன்னோட்டப் பயன்பாடு ஏற்றுக் கொள்ளப்படுமாயின், வாடிக்கையாளர் வேண்டிக் கொள்கிற முழுமையான கணிப்பொறி தரவாக்கப்பணி எடுத்துக் கொள்ளப்படும்.
- வெவ்வேறு வகைப்பட்ட தரவுகளைக் கணிப்பாறித் தரவாக்க, வெவ்வேறு நுட்பங்கள் பயன்படுத்தப்படுகின்றன. மூல ஆவணங்களுடைய கணிப்பொறித் தகவல் வடிவத்தின் தரத்தை மேம்படுத்த பற்பல உயர்நுட்ப மென்பொருள் தொகுப்புகள் கிடைக்கின்றன.
- கணிப்பொறித் தரவாக்கப்பட்ட தகவல்கள் வரிசைப்படுத்தப்பட்டு, எளிதாக அணுகும் வகையில், உள்ளடக்க அட்டவணைகள் தயாரிக்கப்படுகின்றன. கணிப்பொறித் தகவலைச் சேமித்து வைக்க மீவுயர் தொழில்நுட்ப மற்றும் நம்பகத் தன்மையுள்ள சேமிப்பு வசதிகள் பயன்படுத்தப்படுகின்றன.

கணிப்பொறித் தகவலாக்கத்தினால் பல்வேறு பலன்கள் கிடைக்கின்றன. அவற்றுள் முக்கியமான சில பலன்கள் :

- ஆவணங்களை நீண்ட காலம் பாதுகாக்க முடியும்.
- முக்கியமான ஆவணங்களை ஒரே இடத்தில் சேமித்து வைக்க முடியும்.
- தகவல்களை மிக எளிதாக அணுகி, எளிதாகப் பயன்படுத்த முடியும்.
- தேவையான தகவல்களை, குறிப்பாகப் படங்கள் மற்றும் உரைப்பகுதிகளை விரைவாகவும், மிகச் சிறப்பாகவும் தேடிக் கண்டறிய முடியும்.
- படங்கள் மற்றும் உரைத்தகவல்களை மிக எளிதாகப் பரிமாறிக் கொள்ள முடியும்.
- குறுவட்டுகள், இணையம் மற்றும் பிற மின்னணு ஊடகங்கள் மூலமாக மிக எளிதாகத் தகவல் பரிமாற்றம் செய்ய முடியும்.

20.7 வலையகச் சேவைகள்

கீழ்க்காணும் வலையகச் சேவைகளை அணுகிப் பயன்பெறவும் கணிப்பொறிகள் உதவுகின்றன:

- வேளாண்மைச் சந்தைப்படுத்தல் பிணையம்
- வேலைவாய்ப்பு ஆலோசணை
- இணையம்வழி வேலைவாய்ப்பு
- பொதுச் சேமநிதி
- பல்வேறு தேர்வுகளின் முடிவுகள்

உலகின் ஏதோ ஒரு மூலையிலிருந்தும் பயன்படுத்திக் கொள்ளக்கூடிய இன்னும் பல ஐடீஇஎஸ் சேவைகளை வருங்காலத்தில் மிக விரைவிலேயே காண இருக்கிறோம்.

பயற்சிகள்

இந்தப் பாடத்தில் விளக்கப்பட்ட பயன்பாட்டு மென்பொருள்கள் பற்றி மேலும் புரிந்து கொள்ள உதவும் பல்லூடக விளக்கப் படங்கள் தனியாக ஒரு குறுவட்டில் உங்கள் பள்ளிக்குத் தரப்பட்டுள்ளது. அதன் உள்ளடக்கத்தை நீங்கள் கட்டாயம் பார்க்க வேண்டும். இக்குறுவட்டைப் பெற உங்கள் ஆசிரியரை அணுகுக. பாடத்தில் குறிப்பிடடப்பட்டுள்ள இணைய தளங்களையும் பாாவையிடுங்கள்.

அதிகாரம் 21

கணிப்பொறி நன்னெறி (COMPUTER ETHICS)

இரண்டாம் உலகப்போரின் போது **நார்பெர்ட் வெய்னர்** (Norbert Weiener) எழுதிய நூலில் முதன்முதலாகக் கணிப்பொறி நன்னெறிக் கோட்பாடுகள் பற்றிக் குறிப்பிட்டுள்ளார். வெய்னரின் நூலில் குறிப்பிடப்பட்டுள்ளவை :

(1) மனித வாழ்வின் நோக்கம் பற்றிய ஒரு விளக்கம்

(2) நீதிநெறியின் நான்கு கோட்பாடுகள்

(3) நன்னெறிகளை நடைமுறைப்படுத்தவதற்கான ஒரு சக்தி வாய்ந்த வழிமுறை.

(4) கணிப்பொறி நன்னெறி தொடர்பான அடிப்படைக் கேள்விகள் பற்றிய கருத்துரைகள்.

(5) முக்கியமான கணிப்பொறி நன்னெறிக் கோட்பாடுகளுக்குரிய எடுத்துக்காட்டுகள்.

1960களின் இடைப்பகுதியில் கலிஃபோர்னியாவிலுள்ள மென்லோ பார்க்கில் உள்ள **எஸ்ஆர்ஐ இன்டர்நேஷனலைச்** சேர்ந்த **டான்பார்க்கர்** (Donn Parker) கணிப்பொறிப் பணியாளர்கள் கணிப்பொறியைத் தீங்கு எண்ணத்துடன் சட்டப்புறம்மாகப் பயன்படுத்தியதை ஆய்வுசெய்யத் தொடங்கினார். 1980களில் அமெரிக்காவிலும் ஐரோப்பாவிலும் தகவல்தொழில்நுட்பத்தின் எண்ணற்ற சமூக, நன்னெறி தொடர்பான தீங்குகள் மக்கள் பிரச்சனையாக மாறின. கணிப்பொறி மூலமான குற்றங்கள், கணிப்பொறிக் கோளாறுகளினால் ஏற்பட்ட பேரிழப்புகள், கணிப்பொறித் தரவுத் தளங்கள் வழியாக ஒருவரின் தனிமறையில் (privacy) குறுக்கீடுகள், மென்பொருள் சொத்துரிமை தொடர்பான சட்ட வழக்குகள் அவற்றுள் சில.

1990களில் அனேக பல்கலைக் கழகங்கள், கணிப்பொறி நன்னெறி பற்றிய முறைப்படியான படிப்புகளை அறிமுகப்படுத்தின. பற்பல பாடப் புத்தகங்கள் மற்றும் பிற படிப்பு விளக்கவுரைகள் உருவாக்கப்பட்டன. இவை, ஆராய்ச்சிக்கான புதிய களங்கள் உருவாகக் காரணமாயின, ஆய்விதழ்கள் வெளிவருவதற்கு அடித்தளமிட்டன.

பொதுவாகச் சொல்வதெனில், **நன்னெறி** என்பது. ஒழுக்கத்தின் தரப்பாடுகளைத் தீர்மானிக்கிற விதிமுறைகள் அல்லது சமூகம் ஏற்றுக் கொள்ளக்கூடிய நடத்தைகளைக் குறிப்பதாகக் கொள்ளலாம். இன்றைக்குப் பல கணிப்பொறிப் பயனாளர்கள் தகவல் தொழில்நுட்பம் தொடர்பான நடவடிக்கைகளைப் பொருத்தமட்டில் நன்னெறி என்பது எது, எது நன்னெறி இல்லை என்கிற கேள்விகளை எழுப்புகின்றனர். நன்னெறி தொடர்பான சில பொதுவான வழிகாட்டுதல்கள் பயனுள்ளவையாக அமையும் என்பதில் ஐயமில்லை. தகவல் தொழில்நுட்பத்தில் அவற்றைப் பயன்படுத்த ஏதுவாக இருக்கும்.

கணிப்பொறி நன்னெறி பற்றிய பொதுவான வழிகாட்டுதல்கள் இவற்றுக்கெல்லாம் தேவைப்படுகின்றன.

- சொந்தத் தரவுகளைப் பாதுகாத்தல்
- கணிப்பொறிக் குற்றம்
- பாதுகாப்பு அரண்களை உடைத்தல்

21.1 **தரவுப் பாதுகாப்பு (**Data Security)

கீழ்க்காணும் வழிமுறைகள் சிலவற்றைப் பயன்படுத்தி ஒருவரின் சொந்தத் தரவுகளைப் பாதுகாத்துக் கொள்ள முடியும்.

நேரடிப் பாதுகாப்பு (Physical Security):

நேரடிப் பாதுகாப்பு என்பது, கணிப்பொறி மென்பொருள்களைக் குறிப்பாக தரவுகள் சேமிக்கப்பட்டுள்ள மின்காந்த வட்டுகளையும் மற்றும் சட்டப்புறம்பாக அணுகவும் களவாடவும் பழுதாக்கவும் அழிக்கவும் ஏதுவான பிற பொருள்களையும் பாதுகாத்துக் கொள்வதைக் குறிக்கிறது. இத்தகு கணிப்பொறி வளங்களை அணுகுவோர்க்குக் கட்டுப்பாடு விதிப்பதன் மூலம் இப்பாதுகாப்பை வழங்கிட முடியும்.

சொந்தப் பாதுகாப்பு (Personal Security)

சொந்தப் பாதுகாப்பு என்பது, அனுமதி பெற்றவர்கள் மட்டுமே கணிப்பொறி முறைமைக்குள் நுழைய முடியுமாறு மென்பொருளைத் தகவமைப்பதைக் குறிக்கிறது. பயனர் பெயர் (User ID) மற்றும் கடவுச் சொல் (Password) ஆகியவை இத்தகைய பாதுகாப்புக்கான கருவிகளாகும். கணிப்பொறியில் தகவலறியத் தேவை உள்ளவர்கள் மட்டுமே பயனர் பெயரும் கடவுச்சொல்லும் பெற்றிருக்க வேண்டும்.

பணியாளர் பாதுகாப்பு (Personnel Security)

பணியாளர் பாதுகாப்பு என்பது, கணிப்பொறி முறைமையையும், தரவுகளையும் நேர்மையற்ற அல்லது கவனக்குறைவான பணியாளர்களிடமிருந்து பாதுகாப்பதைக் குறிக்கிறது.

21.2 **கணிப்பொறிக் குற்றம் (**Computer Crime)

கணிப்பொறிக் குற்றம் என்பது, கணிப்பொறி மென்பொருள், தரவுகள், இவற்றை அணுகுதல் இவை குற்றத்தின் காரணமாய், நோக்கமாய் அல்லது கருவியாய் அமையக் கூடிய எந்தவொரு சட்டபுறம்பான நடவடிக்கையையும் குறிக்கிறது.

கணிப்பொறிக் குற்றங்கள் சில :

- இணையத்தின் வழியாகப் பணப் பரிமாற்றம் செய்வது தொடர்பான குற்றங்கள்
- கணிப்பொறியைப் பயன்படுத்திச் சட்டப்புறம்பாக வெளிநாட்டுத் தொலைபேசி அழைப்புகளைச் செய்தல்
- ரகசியமான கோப்புகளை அத்துமீறி அணுகுதல்
- வன்பொருள்களைக் களவாடல்
- சொந்தத் தரவுகளை முறைகேடாகப் பயன்படுத்துதல்/ விற்றல்
- வன்பொருள் அல்லது மென்பொருள்களின் உரிமையிலா நகலாக்கம்
- நச்சுநிரல்(virus)
- பாதுகாப்பு அரண் உடைத்தல்(cracking)
- கணிப்பொறி நேரத்தைக் களவாடல்(Theft of Computer time)

ஒட்டுமொத்தக் கணிப்பொறிக் குற்றங்களில் 80% குற்றங்கள் அந்த நிறுவனத்துக்குள்ளேயே நிகழ்பவை என்பதை அறிக. 60% க்கு மேற்பட்ட குற்றங்கள் அறிவிக்கப்படாமலேயே போய்விடுகின்றன.

போலியான வன்பொருள், மென்பொருள்களை உருவாக்குவதும் பயன்படுத்துவதும் **உரிமையிலா நகலாக்கம்** (prïracy) எனப்படுகிறது.

உரிமையிலா நகலாக்கம் செய்ய நாம் முனைவதற்குக் காரணம்

- நாம் இலவசப் பொருள்களை விரும்புகிறோம்.
- நம்மால் இலவசமாகப் பெறமுடியும் என்கிறபோது, எதற்காக விலை கொடுத்து வாங்க வேண்டும் என்கிற எண்ணம்.
- நமது எண்ணமும் செயல்பாடுகளும் சுயநலத்தை அடிப்படையாய்க் கொண்டவை.
- குறைந்த ஆபத்து விளைகிற, பண ரீதியாகப் பயன்தருகிற ஒன்றைப் பெறும் வாய்ப்புக் கிடைக்கும் எனில் அதைத் தயங்காமல் செய்து முடிப்போம். நாம் வாழும் முதலாளித்துவ சமூகம் அவ்வாறு நம்மைப் பழக்கியிருக்கிறது.

நச்சுநிரல் தன்னைத் தானே நகலெடுத்துப் பெருக்கிக்கொள்ளும். உங்கள் கணிப்பொறியில் சேமித்து வைத்துள்ள தரவுகளுக்கும் கோப்புகளுக்கும் பாதிப்பு ஏற்படுத்தக் கூடியது. சவாலைச் சாதித்துக் காட்ட வேண்டும், பிறர் உடைமைகளுக்கு அழிவை ஏற்படுத்த வேண்டும் என்கிற தூண்டுதல் உள்ள, திறன்மிக்க மாபெரும் நிரலாக்கத் திறன்கொண்ட நிரலர்கள் எழுதுகின்ற நிரல்களே இவை. அறியப்பட்ட நச்சுநிரல்கள் 57000க்கும் அதிகமானவை. ஒவ்வொரு நாளும் 6 புதிய நச்சுநிரல்கள் கண்டறியப்படுகின்றன.

நிறுவனங்களில் பயன்படுத்தப்படும் கணிப்பொறிகளில் பெரும்பாலானவை எல்லா நேரமும் பயன்பாட்டில் இருப்பதில்லை. பல மணி நேரம் அவை வாளா இருக்கின்றன. இப்படி இருக்கும் நேரங்களில் கணிப்பொறிகளைப் பலனுள்ள முறையில் பயன்படுத்தப் பல்வேறு வழிமுறைகள் ஆய்வு செய்யப்பட்டு வருகின்றன. எனினும், இவ்வாறு ஒரு நிறுவனத்தில் பயன்படுத்தப்படாமல் இருக்கின்ற கணிப்பொறி நேரங்கள் சட்டப்புறம்பான வழிகளில் களவாடப்படுகின்றன. நிறுவனத்துக்குத் தெரியாமலேயே வாளா இருக்கும் கணிப்பொறிகளில் வேறுபிற மென்பொருள்கள் இயங்கிக் கொண்டிருக்கின்றன. இதனையே ‘கணிப்பொறி நேரம் களவாடப் படல்’ என்கிறோம்.

கணிப்பொறி நன்னெறி தொடர்பாகப் பரவலாக எடுத்தக்காட்டப்படும் குறிப்பு கணிப்பொறி **நன்னெறிக் கழகம்** வரையறத்துள்ள **கணிப்பொறி நன்னெறிக்கான பத்துக் கட்டளைகள்** ஆகும். அவை கீழே தரப்பட்டுள்ளன.

- பிறருக்குத் தீங்கு விளைவிக்குமாறு ஒரு கணிப்பொறியைப் பயன்படுத்தாதீர்கள்.
- பிறரின் கணிப்பொறிப் பணியில் தலையிடாதீர்கள்
- பிறரின் கணிப்பொறிக் கோப்புகளை நோட்டம் விடாதீர்கள்

  கணிப்பெறியை பயன்படுத்தி களவாடாதீர்கள்.

- பொய் சாட்சி உருவாக்க கணிப்பொறியைப் பயன்படுத்தாதீர்கள்

- நீங்கள் விலை கொடுத்து வாங்காத பதிப்புரிமை உள்ள மென்பொருள்களை நகலெடுக்காதீர்கள் அல்லது பயன்படுத்தாதீர்கள்.

- பிறருடைய கணிப்பொறி வளங்களை அனுமதியின்றியோ, உரிய கட்டணமின்றியோ பயன்படுத்தாதீர்கள்

- பிறரின் அறிவுசார் படைப்புகளை முறைகேடாக அபகரித்துக் கொள்ளாதீர்கள்.

- நீங்கள் உருவாக்கும் மென்பொருள் அல்லது நீங்கள் வடிவமைக்கும் முறைமையின் சமூக விளைவுகளைச் சற்றே எண்ணிப் பாருங்கள்

- உடன் வாழும் மனிதர்களின் உணர்வுகளைக் கணக்கில் கொண்டு உரிய மரியாதை வழங்கும் வகையிலேயே எப்போதும் கணிப்பொறியைப் பயன்படுத்துங்கள்.

கணிப்பொறிக் குற்றங்களுக்கென அரசாங்கம் சிறப்பான சட்டங்களை நிறைவேற்ற வேண்டும். இன்றைக்குப் பலநாடுகள் பல்வேறு வகையான சட்டங்களை நிறைவேற்றி, கணிப்பொறிக் குற்றம் புரிவோர்க்குத் தண்டனைகளை வழங்கி வருகின்றன. இந்தியாவுக்கும் கணிப்பொறிக் குற்றங்களைத் தடுக்க மின்வெளிச் சட்டங்களை (cyber Laws) இயற்றியுள்ளது.

21.3 அரண் உடைத்தல் (Cracking)

சட்டப்புறம்பான முறையில் ஒரு கணிப்பொறி முறைமையில் அல்லது பிணையத்தில் நுழைவதையே **அரண் உடைத்தல்** என்கிறோம். கணிப்பொறியில் இருக்கும் சிறப்பான வளங்களைச் சட்டப்புறம்பாகப் பயன்படுத்துவதே அரண் உடைத்தலின் அடிப்படை நோக்கம் ஆகும். வளங்கள் என்பவை வன்பொருள், மென்பொருள், கோப்புகள், முறைமைத் தகவல்கள் இவற்றுள் எதுவாகவும் இருக்கலாம். பழிவாங்கல், வணிகப் போட்டி, சாகசம் ஆகியவை இத்தகைய குற்றத்துக்கான பிற காரணங்கள் ஆகும்.

21.4 வேலை, குடும்பம், பொழுதுபோக்கு

கையகக் கணிப்பொறிகளும் தொலைக் கணிப்பணியும், (Tele Computing) மனிதர்கள் தமது பணிகளை எந்த இடத்துக்கும் எந்த நேரத்திலும் எடுத்துச் சென்று செய்து முடிக்கும் நிலையை உருவாக்கியுள்ளன. இதன் விளைவாக, அவர்களது அலுவலகப் பணிகள் குடும்பத்தினருடன் இருக்கும் நேரத்தையும், விடுமுறைகளையும், ஓய்வு நேரத்தையும் விழுங்கி விடுகின்றன. குடும்பம், நண்பர்கள் போன்ற மரபுவழிப்பட்ட அமைப்புகளையே பலவீனப்படுத்துகின்றன. தனிவாழ்க்கை, பொது வாழ்க்கை என்ற இரண்டுக்கும் இடையேயான கோடு அழிந்து வருகிறது. கணிப்பொறி நன்னெறியில் இது மிகவும் முக்கியமான பிரச்சினையாக உருவெடுத்து வருகிறது.

பயிற்சி வினாக்கள்

1. கணிப்பொறி அமைப்பில் நுழைவதற்குக் கடவுச்சொல் (password) இருக்க வேண்டிய தேவை என்ன ?

2. கணிப்பொறி இயக்க முறைமை (Operating System)பாதுகாப்பினை எவ்வாறு மேம்படுத்துகிறது ?