பாடம் 10

N16RKZ

பாய்வுக் கட்டுப்பாடு

கற்றலின் நோக்கங்கள்

இந்தப் பாடப்பகுதியைக் கற்றபின் மாணவர் அறிந்து கொள்வது.

- பல விதமான கூற்றுகளைப்பற்றி தெரிந்து கொள்ளுதல்
- பல்வேறு பாய்வுக் கட்டுப்பாட்டு கூற்றுகளை உருவாக்குதல்

10.1 அறிமுகம்

C++ நிரலாக்கத்தின் மாறிகள், மாறிலிகள், செயற்குறிகள், தரவினங்கள் போன்றவற்றின் அடிப்படை கருத்துகளைப் பற்றி முந்தைய பாடப்பகுதியில் பார்த்தோம். பொதுவாக ஒரு நிரலின் கூற்றுகள் தொடக்கத்திலிருந்து இறுதி வரை ஒன்றன் பின் ஒன்றாக நிறைவேற்றப்படும். இருப்பினும் இத்தகைய வரிசைமுறை இயக்கம் கட்டுப்பாடுடையதாகவும் குறைந்த பயனுள்ளதாகவும் உள்ளது. பல சூழ்நிலைகளில் குறிப்பிட்ட நிபந்தனையின் 1 பி+அடிப்படையில் நிரல் குறிமுறையின் தொகுதி நிறைவேற்றப்பட வேண்டும் என்று தீர்மானிப்பது அதிக பயனுள்ளதாக இருக்கும். இத்தகைய சூழ்நிலையில், பாய்வுக் கட்டுப்பாடு நிரலின் ஒரு தொகுதியிலிருந்து மற்றொரு பகுதிக்கு தாவச் செய்கிறது. அத்தகைய தாவலுக்கு காரணமான நிரல் கூற்றுகள் கட்டுப்பாடு கூற்றுகள் எனப்படும்.

இந்த பாடப்பகுதியில் கட்டுப்பாட்டு கட்டமைப்பின் அடிப்படை செயல்பாடுகளான தேர்ந்தெடுப்பு, மடக்கு மற்றும் தாவுதல் கூற்றுகள் பற்றி காண்போம்.

10.2 கூற்றுகள் (Statements)

கணிப்பொறி நிரல் என்பது ஒரு குறிப்பிட்ட பணியை நிறைவேற்ற கொடுக்கப்படும் கூற்றுகள் அல்லது கட்டளைகளின் தொகுப்பாகும். இவ்வகை கூற்றுகள், ஒரு குறிப்பிட்ட செயலை நிறைவேற்ற பயன்படுகிறது. செயல் எனப்படுவது, ஒரு மாறி அறிவித்தல், கோவைகளின் மதிப்பீடு, மதிப்பிருத்தல், தீர்மானிப்பு, மடக்கு போன்ற செயல்பாடுகளாக இருக்கலாம்.

C++ நிரலில் இரு வகையான கூற்றுகள் பயன்படுத்தப்படுகின்றன.

(i) வெற்று கூற்று (Null Statements)

(ii) கலவை கூற்று (Compound Statements)

10.2.1 வெற்று கூற்று

அரைப்புள்ளியை மட்டுமே கொண்டிருக்கும் கூற்று "வெற்று அல்லது வெறுமைக் கூற்று" எனப்படும். இதன் பொது வடிவம் பின்வருமாறு:

```
; // இது ஒரு வெற்றுக் கூற்று
```

வெற்றுக் கூற்றுகளை, மடக்கில் இட உணர்த்திகளாக பயன்படுத்தலாம் அல்லது கலவை கூற்றுகள் அல்லது செயற்கூறுகளின் முடிவில் இடம் பெறச்செய்யலாம்.

10.2.2 கலவை (தொகுதி) கூற்று

C++ கூற்றுகளின் தொகுப்பினை நெளிவு அடைப்புக்குறிகளுக்குள் உள்ளடக்க அனுமதிக்கிறது. இந்த கூற்றுகளின் தொகுப்பினை கலவை கூற்று அல்லது தொகுதி என்கிறோம்.

கலவை கூற்றின் பொது வடிவம்

```
{
        கூற்று1;
        கூற்று 2;
        கூற்று 3;
}
```

எடுத்துக்காட்டு

```
{
        int x, y;
        x = 10;
        y = x + 10;
}
```

கலவை கூற்று அல்லது தொகுதி ஒற்றை அலகாக கருதப்படுகிறது. இவை நிரலின் எந்த பகுதியில் வேண்டுமானாலும் இடம் பெறலாம்.

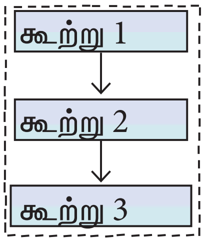
10.3 கட்டுப்பாட்டு கூற்றுகள்

கட்டுப்பாட்டு கூற்றுகள், கட்டளைகளின் பாய்வு வரிசை முறையை மாற்றி அமைக்கும். ஒரு நிரலிலுள்ள கூற்றுகள், வரிசைமுறை, தேர்ந்தெடுப்பு மற்றும் மடக்கு போன்ற கட்டுப்பாட்டு கூற்றுகளாக இயக்கப்படுகிறது. அனைத்து நிரலாக்க மொழிகளும், வரிசைமுறை, தேர்ந்தெடுப்பு மற்றும் மடக்கு கூற்றுகளை கொண்டுள்ளது.

கூற்றுகள் ஒன்றன் பின் ஒன்றாய் வரிசைமுறையில் நிறைவேற்றப்பட்டால், இந்த பாய்வை வரிசைமுறைப் பாய்வு என்கிறோம்.

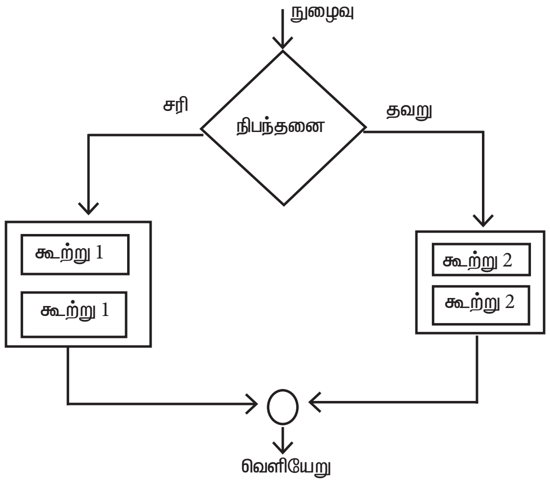
சில சூழ்நிலைகளில், கிளை பிரித்தல், மடக்கு, தாவுதல் மற்றும் செயற்கூறு அழைப்பு போன்ற கூற்றுகள் பாய்வின் ஓட்டத்தை மாற்றியமைக்கும். இந்த செயல்முறையை பாய்வுக் கட்டுப்பாடு (Control Flow) என்கிறோம்.

வரிசைமுறை கூற்று



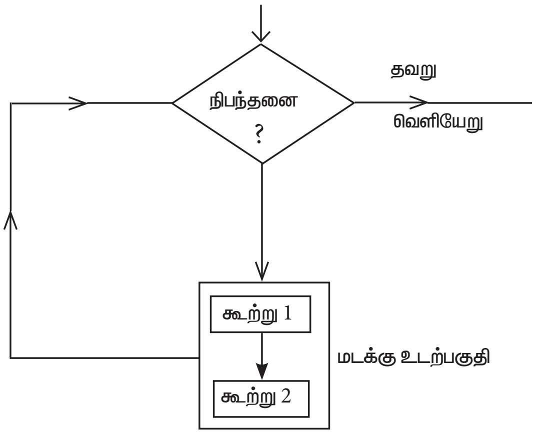
வரிசைமுறை கூற்றுகள் என்பது மேலிருந்து கீழாக ஒன்றன் பின் ஒன்றாக நிறைவேற்றப்படும் கூற்றுகளாகும். இத்தகைய கூற்றுகள் பாய்வு ஓட்டத்தை மாற்றி அமைக்காது. இவை எப்பொழுதும் அரைப்புள்ளியுடன் (;) முற்றுப்பெறுகிறது.

தேர்ந்தெடுப்புக் கூற்று



நிபந்தனை அடிப்படையில் நிறைவேற்றப்படும் கூற்றுகள் தேர்ந்தெடுப்பு கூற்றுகள் எனப்படும். கொடுக்கப்பட்ட நிபந்தனை சரி எனில் சரி கட்டளைத் தொகுதி (கூற்றுகளின் தொகுப்பு) இயக்கப்படும், இல்லையெனில் தவறு கட்டளைத் தொகுதி இயக்கப்படும். நிறைவேற்றப்பட வேண்டிய கூற்றுகளின் தொகுப்பை தீர்மானிக்க உதவுவதால், இக்கூற்றினை தீர்மானிப்புக் கூற்று அல்லது தேர்ந்தெடுப்பு கூற்று எனலாம்.

மடக்குக் கூற்று



மடக்குக் கூற்று என்பது ஒரு கட்டளைத் தொகுதியை நிபந்தனை அடிப்படையில் மீண்டும் மீண்டும் செயல்படுத்தும். கொடுக்கப்பட்ட நிபந்தனை சரி என இருக்கும் வரை, கட்டளைத் தொகுதி மீண்டும் மீண்டும் நிறைவேற்றப்படும். நிபந்தனை தவறாகும் போது தொடர்ந்து இயக்கப்படுவது நிறுத்தப்படுகிறது. இதனை மடக்குக் கூற்று அல்லது பன்முறைச் செயல் கூற்று என்கிறோம். மேலே கொடுக்கப்பட்டுள்ள பாய்வுப் படம் பன்முறைச் செயல் கூற்றினை விளக்குகிறது.

மீண்டும் மீண்டும் இயக்கப்படும் கட்டளைத் தொகுதியினை மடக்கின் உடற்பகுதி என்கிறோம். ஒரு மடக்கில் உள் நுழைய அல்லது வெளியேற காரணமான நிபந்தனையை வெளியேறல் நிபந்தனை அல்லது பரிசோதிப்பு நிபந்தனை என்கிறோம். பொதுவாக, அனைத்து வகை நிரலாக்க மொழிகளுக்கு சிக்கல்களின் தேவைக்கேற்ப இவ்வகை கூற்றினை நிரல் குறிமுறை எழுத பயன்படுத்திக் கொள்கிறது. C++ நிரலாக்க மொழியும் இவ்வகை கூற்றுகளை பயன்படுத்திக் கொள்கிறது. தேர்ந்தெடுப்புக் கூற்று மற்றும் மடக்குக் கூற்றுகள் கொடுக்கப்படும் நிபந்தனை கோவையின் வெளியீட்டு மதிப்பை அடிப்படையாக கொண்டு இயங்குகிறது. நிபந்தனை கோவை சரி அல்லது தவறு என்ற மதிப்புகளை விடையாக தரும்.

Chapter 10.indd 156 4/13/2020 6:10:02 PM

குறிப்பு

குறிப்பு:

C++ மொழியில், சுழியம் அல்லாத எந்தவொரு எண்ணும் "சரி" என எடுத்துக் கொள்ளப்படும், (எதிர்மறை எண் உள்பட) மற்றும் சுழியம் "தவறு" என்று எடுத்துக் கொள்ளப்படும்.

10.4 தேர்ந்தெடுப்பு கூற்றுகள்

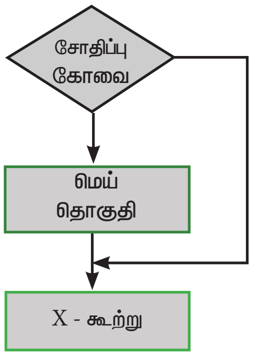
ஒரு நிரலில் தீர்மானிப்புக் கூற்று ஒரு பகுதியில்லிருந்து மற்றோரு பகுதிக்கு கட்டுப்பாட்டை ஒரு முறைத் தாவச்செய்கிறது. இதை பல வழிகளில் மேற்க்கொள்ளலாம் முக்கியமாக **if..else** கூற்று பயன்படுகிறது. இரண்டு குற்றுகளுகக்கிடையே ஒன்றை தேர்வு செய்கிறது மற்றொரு தீர்மானிப்புக் கூற்று **switch** என்பது ஒரு மாறியின் அடிப்படையில் கிளைப்பிரிப்பு கூற்று தொகுதிகளை உருவாக்கும்

10.4.1 if கூற்று

if கூற்று என்பது ஒரு தீர்மானிப்புக் கூற்றாகும். if கூற்று நிபந்தனையை சோதித்து, நிபந்தனை சரி எனில் சரித்தொகுதி கூற்று அல்லது கட்டளைத்தொகுதி இயக்கப்படும். இல்லையெனில் சரித் தொகுதி தவிர்க்கப்படும். if கூற்றின் தொடரியல்.

```
if (நிபந்தனை கோவை)
        மெய் தொகுதி;
x-கூற்று
```

மேற்கண்ட கட்டளை அமைப்பில், if என்ற சிறப்புச்சொல் கோவை அல்லது நிபந்தனையை அடைப்புக்குறிக்குள் கொண்டிருக்க வேண்டும். கொடுக்கப்பட்ட கோவையின் மதிப்பு சரி எனில் (0 சுழியம் அல்லாத) சரித்தொகுதி இயக்கப்படும். அதைத் தொடர்ந்து x-கூற்று எடுத்துச் செல்லும். ஒரு சரித் தொகுதி ஒற்றை கூற்று, கலவை கூற்று அல்லது வெற்று கூற்றினை கொண்டிருக்கலாம்.



எடுத்துக்காட்டு 10.1 if கூற்றைப் பயன்படுத்தி ஒரு நபர் வாக்களிக்க தகுதியானவரா என சோதிக்க C++ நிரல் ஒன்று எழுதுக.

```
#include <iostream>
using namespace std;
int main()
{
        int age;
        cout << "\n Enter your age: ";
        cin >> age;
        if (age >= 18)
                cout<< "\n You are eligible for voting ....";
        cout << "This statement is always executed.";
        return 0;
}
```

if கூற்று, அதற்கு கீழே கொடுக்கப்படும் ஒரே ஒரு நிரல் கூற்றை மட்டுமே இயக்கும். எனவே, நெளிவு அடைப்புக் குறியீடு பயன்படுத்த வேண்டிய அவசியமில்லை.

வெளியீடு

```
Enter your age: 23
You are eligible for voting….
This statement is always executed.
```

10.4.2 if-else கூற்று

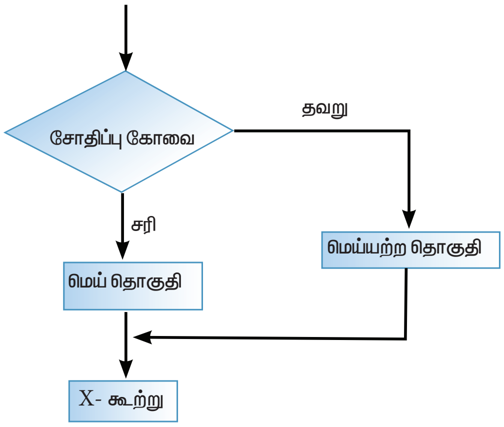
ஒரு if else கூற்று என்பதும் கட்டுப்பாட்டு பாய்வு கூற்றாகும். இது நிபந்தனையின் அடிப்படையில் ஒரு செயலை செய்யப் பயன்படுகிறது. If else கூற்றில், முதலில் கோவை அல்லது நிபந்தனை சரியா அல்லது தவறா என மதிப்பிடப்படுகிறது. கிடைக்கப்பெற்ற விடை சரியெனில், சரித்தொகுதியுனுள் இடம் பெறும்

Chapter 10.indd 157 4/13/2020 6:10:02 PM

கூற்றுகள் இயக்கப்படும். தவறு தொகுதி நிறைவேற்றப்படாது, கிடைக்கபெற்ற விடை தவறு எனில், தவறு தொகுதியினுள் இடம் பெறும் கூற்றுகள் இயக்கப்படும், சரித்தொகுதி நிறைவேற்றப்படாது. If else கூற்றின் கட்டளை தொடரியல் பின்வருமாறு:

```
if (கோவை)
{
        மெய் தொகுதி;
}
else
{
        மெய்யற்ற தொகுதி;
}
x-கூற்று
```

மேற்கண்ட கட்டளை அமைப்பில், if கூற்றுடன் கொடுக்கப்பட்ட நிபந்தனை முதலில் மதிப்பீடு செய்யப்படுகிறது. நிபந்தனை சரி எனில், சரித்தொகுதி இயக்கப்பட்டு, பின் கட்டுப்பாடு X கூற்றுக்கு இட்டுச் செல்கிறது,. நிபந்தனை தவறு எனில், தவறு தொகுதி இயக்கப்பட்டு, பின் கட்டப்பாடு X கூற்றுக்கு அனுப்பிவைக்கப்படுகிறது.



எடுத்துக்காட்டு 10.2 if-else கூற்றைப் பயன்படுத்தி கொடுக்கப்பட்ட எண் ஒற்றைப் படை அல்லது இரட்டைப் படை எண்ணா எனக் காணும் C++ நிரல் ஒன்றை எழுதுக.

```
#include <iostream>
using namespace std;
int main()
{
        int num, rem;
        cout<< "\n Enter a number: ";
        cin>>num;
        rem = num % 2;
        if (rem==0)
                cout<< "\n The given number" <<num<< " is Even";
        else
                cout<< "\n The given number "<<num<< " is Odd";
        return 0;
}
```

வெளியீடு

```
Enter number: 10
The given number 10 is Even
```

மேற்கண்ட நிரலில், கொடுக்கப்பட்ட எண்ணின் வகுமீதி rem என்ற மாறியில் இருத்தப்படும். rem ன் மதிப்பு சுழியம் எனில், கொடுக்கப்பட்ட எண் இரட்டைப் படை எண் (even number) என்ற வெளியீடும், இல்லையெனில், ஒற்றைப் படைஎண் (odd number) என்ற வெளியீடும் கிடைக்கும்.

10.4.3 பின்னலான if கூற்று:

ஒரு if கூற்று, மற்றொரு if கூற்றை கொண்டிருந்தால் அது பின்னலான if கூற்று என்று அழைக்கப்படுகிறது. பின்னலான if கூற்றுகள், பின்வரும் மூன்று வடிவங்களில் அமைக்கப்படும். அவையாவன

(1) ஒரு if கூற்றுக்கு உள்ளே மற்றொரு if கூற்றை அமைத்தல்

(2) ஒரு if கூற்றின் else கூற்றுக்குள் மற்றொரு if கூற்றை அமைத்தல்

(3) ஒரு if கூற்றுக்கு உள்ளே ஒரு if கூற்றும், else கூற்றுக்கு உள்ளே மற்றொரு if கூற்றும் அமைத்தல்.

இந்த மூன்று வடிவங்களின் தொடரியல் கீழே கொடுக்கப்பட்டுள்ளது.

Chapter 10.indd 158 4/13/2020 6:10:02 PM

ஒரு if கூற்றுக்கு உள்ளே மற்றொரு if கூற்றை அமைத்தல்

```
if (கோவை-1)
{
        if (கோவை - 2)
        {
          மெய் பகுதி கூற்றுகள்;
        }
        else
        {
         மெய்யற்ற பகுதி கூற்றுகள்;
        }
}
    else பகுதியின் உடற்பகுதி;
```

ஒரு if கூற்றின் else கூற்றுக்குள் மற்றொரு if கூற்றை அமைத்தல்

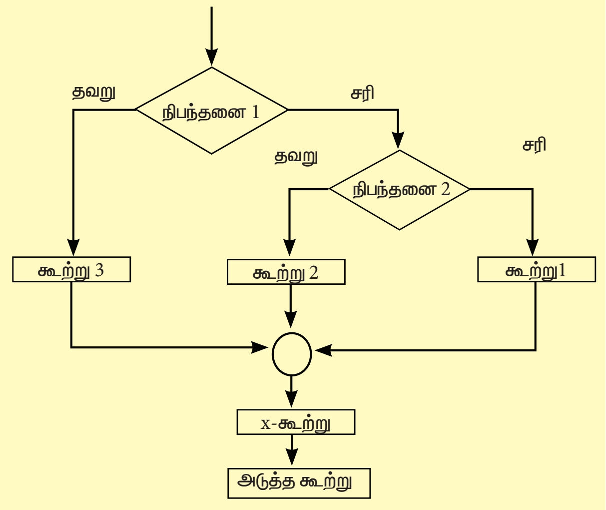
```
if (கோவை-1)
{
        மெய் பகுதியின் உடற்பகுதி;
}
else
{
        if (கோவை - 2)
        {
          மெய் பகுதி கூற்றுகள்;
        }
        else
        {
         மெய்யற்ற பகுதி கூற்றுகள்;
        }
}
```

ஒரு if கூற்றுக்கு உள்ளே ஒரு if கூற்றும், else கூற்றுக்கு உள்ளே மற்றொரு if கூற்றும் அமைத்தல்.

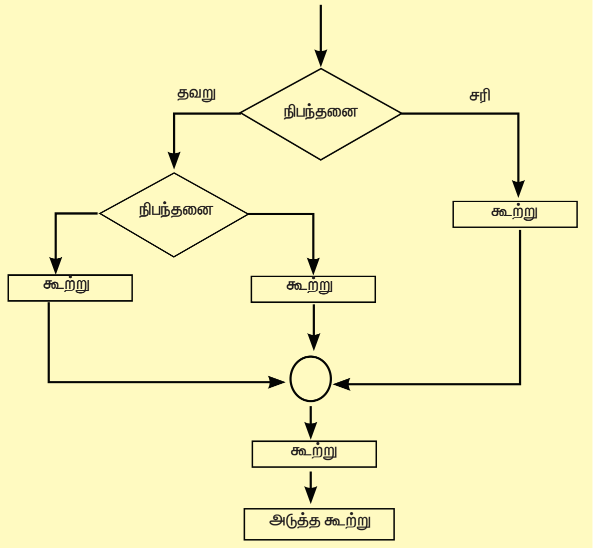
```
if (கோவை)
{
        if (கோவை)
        {
          மெய் பகுதி கூற்றுகள்;
        }
        else
        {
          மெய்யற்ற பகுதி கூற்றுகள்;
        }
}
else
{
        if (கோவை)
        {
          மெய் பகுதி கூற்றுகள்;
        }
        else
        {
          மெய்யற்ற பகுதி கூற்றுகள்;
        }
}
```

மேலே கொடுக்கப்பட்டுள்ள முதலாவது தொடரியல், முதன்மை if கூற்றுக்குள் கொடுக்கப்பட்டுள்ள கோவை-1 முதலில் மதிப்பிடப்படும். அதன் மதிப்பு சரி எனில், அதனுள் கொடுக்கப்பட்டுள்ள பின்னலான if கூற்றுக்கு கட்டுபாட்டு பாய்வு செல்லும். பின்னலான if கூற்றின் நிபந்தனை சரி எனின், அதனுள் கொடுக்கப்பட்ட சரி தொகுதியின் கூற்றும், தவறு எனில் else கூற்றினுள் கொடுக்கப்பட்டுள்ள தவறு தொகுதியின் கூற்றும் இயக்கப்படும். முதன்மை if கூற்றின் நிபந்தனை கோவை மதிப்பு தவறு எனில், முதன்மை else தொகுதிக்குள் கொடுக்கப்பட்டுள்ள நிரல் கூற்றுகள் இயங்கும்.

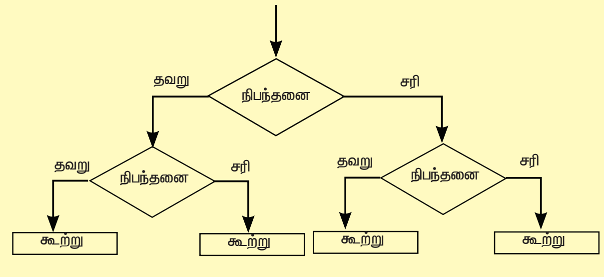
மேலே கொடுக்கப்பட்டுள்ள மூன்று வகையான பின்னல் if கூற்றுகளின் பாய்வுப் படங்கள் கீழே கொடுக்கப்பட்டுள்ளது.



பாய்வுப் படம் 10.1 ஒரு if கூற்றுக்கு உள்ளே மற்றொரு if கூற்றை அமைத்தல்



பாய்வுப் படம் 10.2 ஒரு if கூற்றின் else கூற்றுக்குள் மற்றொரு if கூற்றை அமைத்தல்



பாய்வுப் படம் 10.3 ஒரு if கூற்றுக்கு உள்ளே ஒரு if கூற்றும், else கூற்றுக்கு உள்ளே மற்றொரு if கூற்றும் அமைத்தல்.

எடுத்துக்காட்டு 10.3 பின்னலான if கூற்றினை பயன்படுத்தி தரநிலைக்கு ஏற்ப விற்பனை தரகை (Commission) கணக்கிட நிரல் ஒன்று எழுது.

```
#include <iostream>
using namespace std;
int main()
{
        int sales, commission;
        char grade;
        cout << "\n Enter Sales amount: ";
        cin >> sales;
        cout << "\n Enter Grade: ";
        cin >> grade;
        if (sales > 5000)
        {
                commission = sales * 0.10;
                cout << "\n Commission: " << commission;
        }
        else
        {
                commission = sales * 0.05;
                cout << "\n Commission: " << commission;
        }
        cout << "\n Good Job ..... ";
        return 0;
}
```

வெளியீடு:

```
Enter Sales amount: 6000
Enter Grade: A
Commission: 600
Good Job .....
```

10.4.4 if -else-if அடுக்கு

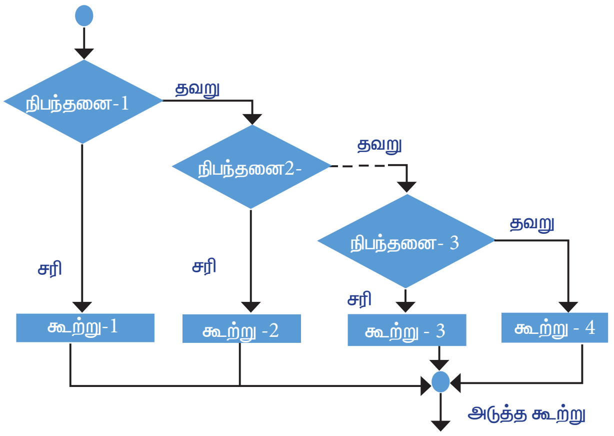
if-else அடுக்கு என்பது பல வழி தீர்மானிப்பு கூற்றாகும். இந்த வகையான கூற்றில், if என்னும் சிறப்புச் சொல்லைத் தொடர்ந்து ஒன்று அல்லது ஒன்றுக்கு மேற்பட்ட else if கூற்றுகள் இடம் பெற்றிருக்கும். இறுதியாக else கூற்றுடன் முடிவு பெறும்.

if-else அடுக்கின் தொடரியல்:

Chapter 10.indd 160 4/13/2020 6:10:02 PM

```
if (கோவை-1)
{
  கூற்று-1
}
else
        if(கோவை- 2)
        {
                கூற்று-2
        }
        else
                if (கோவை-3)
                {
                        கூற்று-3
                }
                else
                {
                        கூற்று-4
                }
```

கொடுக்கப்பட்டுள்ள கோவை (Expressions) சரி எனில், அந்த தொகுதியுடன் தொடர்புடைய கூற்று நிறைவேற்றப்படும், அந்த அடுக்கில் இடம் பெற்றிருக்கும் மற்ற கூற்றுகள் தவிர்க்கப்பட்டு விடும். எல்லா கோவையும் தவறு எனில் இறுதியான else கூற்று நிறைவேற்றப்படும்



பாய்வுப் படம் 10.4 if-else அடுக்கு

எடுத்துக்காட்டு 10.4 if-else அடுக்கினை பயன்படுத்தி உன்னுடைய தரநிலைக் கண்டறியும் C++ நிரல் எழுதுக.

```
#include <iostream>
using namespace std;
int main ()
{
int marks;
cout<<" Enter the Marks  :";
cin>>marks;
if( marks >= 60 )
        cout<< "Your grade is 1st class !!" <<endl;
                else if( marks >= 50 && marks < 60)
                        cout<< "your grade is 2nd class !!" <<endl;
                                else if( marks >= 40 && marks < 50)
                                        cout<< "your grade is 3rd class !!" <<endl;
else
        cout<< "You are fail !!" <<endl;
return 0;
}
```

வெளியீடு

```
        Enter the Marks  :60
        Your grade is 1st class !!
```

மதிப்பெண்கள் 60ஐ விட அதிகமானதாகவோ அல்லது சமமானதாகவோ இருப்பின் "Your grade is 1st class !!" என்ற செய்தி வெளியிடப்படும். அடுக்கில் இடம் பெற்ற மற்றவை தவிர்க்கப்பட்டு விடும். மதிப்பெண்கள் 50க்கும், 59க்கும் இடையே இருக்குமானால் "Your grade is 2nd class !!" என்று வெளியிடப்படும், மற்ற அடுக்கு தவிர்க்கப்பட்டு விடும். மதிப்பெண்கள் 40 க்கும் 49க்கும் இடையில் இருந்தால் செய்தி "Your grade is 3rd class !!" என்று வெளியிடப்படும் இல்லையெனில் "You are fail !!" என்ற செய்தி காண்பிக்கப்படும்.

Chapter 10.indd 161 4/13/2020 6:10:02 PM

10.4.5 if else க்கு மாற்றான நிபந்தனை செயற்குறி

நிபந்தனை செயற்குறி அல்லது மும்ம செயற்குறி என்பது if else கூற்றின் மாற்று வழியாகும். நிபந்தனை செயற்குறி ?: என்ற இரண்டு குறியீடுகளை கொண்டுள்ளது. இது மூன்று செயலுருப்புகளை ஏற்கும். நிபந்தனை செயற்குறியின் பாய்வுக்கட்டுப்பாடு கீழே கொடுக்கப்பட்டுள்ளது,

நிபந்தனை செயற்குறியின் தொடரியல்

கோவை 1 ? கோவை 2 : கோவை 3 ;

மேற்கண்ட தொடரியலில், கோவை 1 என்பது மதிப்பிட வேண்டிய நிபந்தனையை குறிக்கும். நிபந்தனை சரி எனில், கட்டுப்பாடு கோவை 2க்குச் செல்லும். இல்லையெனில் கட்டுப்பாடு கோவை 3க்கு எடுத்துச்செல்லப்படும்.

எடுத்துக்காட்டு 10.5 கொடுக்கப்பட்ட இரண்டு எண்களில் எது பெரியது என கண்டறிய நிபந்தனை செயற்குறியை பயன்படுத்தி ஒரு நிரல் எழுதுக.

```
#include <iostream>
using namespace std;
int main()
{
        int a, b, largest;
        cout << "\n Enter any two numbers: ";
        cin >> a >> b;
        largest = (a>b)? a : b;
        cout << "\n Largest number : " << largest;
        return 0;
}
```

வெளியீடு:

```
Enter any two numbers: 12 98
Largest number : 98
```

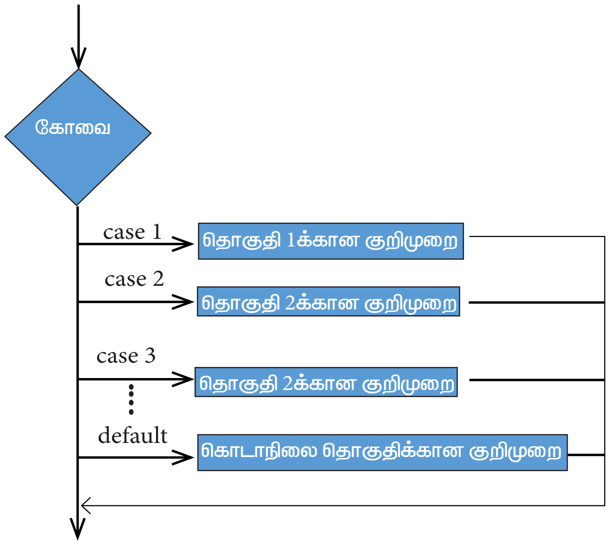
10.4.6 switch கூற்று

switch கூற்று என்பது ஒரு பல வழி கிளைப்பிரிப்பு கூற்றாகும். இது கோவையின் மதிப்பின் அடிப்படையில் பல்வேறு நிரல் பகுதிகளுக்கு கட்டுப்பாட்டை எடுத்துச் செல்வதற்கு எளிதாக வகை செய்கிறது. switch கூற்று ஒன்றுக்கும் மேற்பட்ட if else கூற்றுகளுக்கு மாற்றாக அமைந்துள்ளது.

switch கூற்றின் தொடரியல்;

```
switch(கோவை)
{
        case constant 1:
                கூற்று(s);
                break;
        case constant 2:
                கூற்று(s);
                break;
        .
        .
        .
        .
        default:
                கூற்று (s);
}
```

மேற்கண்ட தொடரியலில், கோவை மதிப்பீடு செய்யப்பட்டு, பெறப்பட்ட மதிப்பு ஏதேனும் ஒரு case கூற்றில் கொடுக்கப்பட்ட மாறிலியின் மதிப்புடன் பொருந்தியிருப்பின், அதனுடன் தொடர்புடைய கட்டளைத் தொகுப்பு நிறைவேற்றப்படும். இல்லையெனில் தேர்வுக்கு கீழே உள்ள கூற்றுகள் நிறைவேற்றப்படும். switch கூற்றின் இயங்கு முறை பாய்வுப்படமும் படம் 10.5ல் கொடுக்கப்பட்டுள்ளது.



படம் 10.5 switch கூற்றின் இயங்குமுறை

விதிமுறைகள்:

1. switch கூற்றில் தரப்பட்டுள்ள கோவையின் விடை மாறிலி மதிப்பாக இருத்தல் வேண்டும் இல்லையெனில அது ஒரு பிழையான வடிவமாகும்.

Chapter 10.indd 162 4/13/2020 6:10:03 PM

2. ஒரே மதிப்பு இரண்டு case கூற்றுகளில் அனுமதிக்கப்பட மாட்டாது.
3. default கூற்று கட்டாயமில்லை.
4. switch கூற்றின் நிரல் தொகுதியை நிறைவு செய்ய break கூற்று பயன்படுகிறது. நிரல் பாய்வு break கூற்றை அடையும் போது, switch கூற்றிலிருந்து வெளியேறி, switch கூற்றை தொடர்ந்து கொடுக்கப்பட்டுள்ள நிரல் கூற்றுகளை இயக்கும்.
5. break கூற்று கொடுப்பது கட்டாயமில்லை. break கூற்று கொடுக்கப்படவில்லை எனில், case நிபந்தனை நிறைவேற்றப்படும் நிரல் கூற்றை தொடர்ந்து உள்ள அனைத்து கூற்றுகளும் கட்டுப்பாடின்றி இயங்கும்.
6. பின்னலான switch கூற்றுகளும் அமைக்கப்படும்.

எடுத்துக்காட்டு 10.6 – switch கூற்றை விளக்கும் C++ நிரல்

```
#include <iostream>
using namespace std;
int main()
{
        int num;
        cout << "\n Enter week day number: ";
        cin >> num;
        switch (num)
        {
                case 1 : cout << "\n Sunday"; break;
                case 2 : cout << "\n Monday"; break;
                case 3 : cout << "\n Tuesday"; break;
                case 4 : cout << "\n Wednessday"; break;
                case 5 : cout << "\n Thursday"; break;
                case 6 : cout << "\n Friday"; break;
                case 7 : cout << "\n Saturday"; break;
                default: cout << "\n Wrong input....";
        }
}
```

வெளியீடு:

```
        Enter week day number: 6
        Friday
```

10.4.7 switch எதிர் if-else : சில வேறுபாடுகள்

"if-else" மற்றும் "switch" ஆகிய இரண்டும் தேர்ந்தெடுப்பு கூற்றுகள். தேர்ந்தெடுப்பு கூற்றுகள், நிபந்தனை சரி அல்லது தவறு என்ற அடிப்படையில், நிரலின் கட்டுப்பாட்டை குறிப்பிட்ட கட்டளைத் தொகுதிக்கு எடுத்துச் செல்கிறது. எனினும், அவை செயல்படும் முறைகளில் சில வேறுபாடுகள் உள்ளன. அவை பின்வருமாறு

if-else மற்றும் switch கூற்றின் இடையே உள்ள முக்கிய வேறுபாடுகள்

if-else மற்றும்/ switch கூற்றுகளுக்கிடையான முக்கிய வேறுபாடுகள்:

if-else	switch
If...else கூற்று, நிபந்தனைக் கூற்றின் அடிப்படையில், if தொகுதியில் கொடுக்கப்பட்ட கூற்றுகளை செயல்படுத்த வேண்டுமா, அல்லது else தொகுதியில் கொடுக்கப்பட்ட கூற்றுகளை செயல்படுத்த வேண்டுமா என்பதை தீர்மானிக்கும்.	switchல் கொடுக்கப்பட்ட கூற்று, எந்த நிகழ்வை (case) செயல்படுத்த வேண்டும் என்பதை தீர்மானிக்கும்.
ஒரு if... else கூற்று ஒன்றுக்கும் மேற்பட்ட தேர்ந்தெடுப்புக் கூற்றுகளுக்கு, ஒன்றுக்கும் மேற்பட்ட நிப்ந்தனைக கூற்றுகளை பயன்படுத்தும்	switch கூற்று, ஒன்றுக்கும் மேற்பட்ட தேர்ந்தெடுப்புக் கூற்றுகளுக்கு, ஒற்றை நிபந்தனைக் கூற்றை மட்டுமே பயன்படுத்தும்

if...else கூற்று, கொடுக்கப்பட்ட நிபந்தனைக் கூற்றுகளின் நிகர் நிலையையும், தருக்க நிலையும் சோதிக்கும்	switch கூற்று கொடுக்கப்பட்ட நிபந்தனைகளின் நிகர் நிலையை மட்டுமே சோதிக்கும்
If கூற்று, முழு எண், எழுத்துரு, மிதப்புப் புள்ளி அல்லது பூலியன் தரவு வகைகளை மதிப்பீட்டுக்கு எடுத்துக்கொள்ளும்.	switch கூற்று எழுத்துரு அல்லது முழு எண் தரவு வகைகளை மட்டுமே மதிப்பீட்டுக்கு எடுத்துக்கொள்ளும்.
கொடுக்கப்பட்ட நிபந்தனையின் முடிவு, பொய் (false) என இருப்பின், else தொகுதியில் கொடுக்கப்பட்ட கூற்றுகள் செயல்படுத்தப்படும்.	கொடுக்கப்பட்ட நிபந்தனையின் முடிவு, பொய் (false) என இருப்பின், defaultக்குள் கொடுக்கப்பட்ட கூற்றுகள் செயல்படுத்தப்படும்.

switch கூற்றை விட if கூற்று மிகுந்த எளிமையானது

10.5 பன்முறைச் செயல் (அல்லது) மடக்கு

பன்முறைச் செயல் அல்லது மடக்கு என்பது ஒன்று அல்லது ஒன்றுக்கு மேற்பட்ட கூற்றுகள், நிபந்தனை நிறைவேறும் வரை திரும்பத் திரும்ப இயக்கப்படுவதை குறிக்கும். இந்த கூற்றுகளை பாய்வுக் கட்டுப்பாட்டு கூற்றுகள் எனவும் கூறலாம். இது நிரலின் நீளத்தை குறைக்கவும், நேரத்தை குறைக்கவும், நிரல் இயக்கவும், குறைவான நினைவக இடத்தை எடுத்துக் கொள்ளவும் பயன்படுகிறது. C++ மூன்று வகையான மடக்கு கூற்றுகளை ஏற்கிறது.

- for கூற்று
- while கூற்று
- do-while கூற்று

அனைத்து மடக்கு கூற்றுகளும் குறிப்பிட்ட நிபந்தனை சரி என இருக்கும் வரையில் கட்டளைத் தொகுதியை மீண்டும் மீண்டும் நிறைவேற்றுகிறது. குறிப்பிட்ட நிபந்தனை மடக்கு கட்டுப்பாடு எனப்படுகின்றது. மூன்று மடக்கு கூற்றுகளுக்கும், சுழியம் அல்லாத எந்தவோர் எண்ணும் சரி என எடுத்துக் கொள்ளப்படும் மற்றும் சுழியம் என்பது தவறு என எடுத்துக்கொள்ளலாம்.

மடக்கின் பகுதிகள்

ஒவ்வொரு மடக்கும் பல்வேறு பயன் பாடுகளுக்கான நான்கு கூறுகளை கொண்டுள்ளது.

- தொடக்க மதிப்பிருத்தும் கோவை
- சோதிப்பு நிபந்தனை கோவை
- புதுப்பித்தல் / மிகுத்தல் கோவை
- மடக்கின் உடற்பகுதி

தொடக்க மதிப்பிருத்தும் கோவைகள் : நிரலின் கட்டுப்பாடு மடக்கினுள் நுழையும் முன் கட்டுப்பாட்டு மாறிகள் தொடக்க மதிப்பிருத்தல் வேண்டும். ஒரு கட்டுப்பாட்டு மாறிக்கு தொடக்க மதிப்பிருத்தும் செயல்பாடு தொடக்க மதிப்பிருத்தும் கோவையின் கீழ் இடம் பெறும். கட்டுப்பாட்டு மாறியில் தொடக்க மதிப்பிருத்தல் மடக்கு செயல்படும் முன்பாக, ஒரே ஒரு முறை தான் செயல்படுத்தப்படும்.

பரிசோதிப்பு நிபந்தனை கோவை: பரிசோதிப்பு நிபந்தனை கோவை என்பது ஒரு மடக்கின் உள்ளே அமைந்துள்ள உடற்பகுதியிலிருக்கும் கட்டளைத் தொகுதி நிறைவேற்றப்பட வேண்டுமா இல்லையா என்பதை தீர்மானிக்கும் கோவை அல்லது நிபந்தனையாகும்.

கொடுக்கப்பட்ட நிபந்தனை சரி (அதாவது 1) எனில், மடக்கின் உடற்பகுதி செயல்படுத்தப்படும் இல்லையேல், மடக்கை விட்டு வெளியேறும்.

பரிசோதிப்பு நிபந்தனை கோவையின் சோதிப்பு நிலையை அடிப்படையாகக் கொண்டு, மடக்குகளை நுழைவு சோதிப்பு மடக்கு மற்றும் வெளியேறல் சோதிப்பு மடக்கு என இரண்டு வகையாக பிரிக்கப்படுகிறது.

நுழைவு சோதிப்பு மடக்கில், நிபந்தனை கோவை மடக்கினுள் நுழையும் முன் சோதிக்கப்படுகிறது. அவ்வாறின்றி வெளியேறல் சோதிப்பு மடக்கில் நிபந்தனைக் கோவை மடக்கினை விட்டு வெளியேறும் முன்பாக சோதிக்கப்படுகிறது.

புதுபித்தல் / மிகுத்தல் கோவை: இது மடக்கு கட்டுப்பாட்டு மாறியின் மதிப்பை மாற்றம் செய்வதற்கு பயன்படுகிறது. இந்த கூற்று, மடக்கின் உடற்பகுதி செயல்பட்டு முடிந்த பின், மடக்கின் இறுதியில் செயல்படுத்தப்படுகிறது.

மடக்கின் உடற்பகுதி: மடக்கின் உடற்பகுதி என்பது மீண்டும் மீண்டும் நிறைவேற்றப்பட வேண்டிய கூற்று அல்லது கட்டளைத் தொகுதியை குறிக்கும். நுழைவு சோதிப்பு மடக்கில், முதலில் நிபந்தனை கோவை செயல்படுத்தப்பட்டு, சுழியம் அல்லாத எண்ணாக இருப்பின், மடக்கின் உடற்பகுதி செயல்படுத்தப்படும் இல்லையெனில் மடக்கு முடிவு பெறும். வெளியேறல் சோதிப்பு மடக்கில், முதலில் மடக்கின் உடற்பகுதி செயல்படுத்தப்பட்டு, பின் நிபந்தனைக் கோவை சரிபார்க்கப்படுகிறது. சோதிப்புக் கோவை சரி எனில், மடக்கின் உடற்பகுதி திரும்ப நிறைவேற்றப்படும் இல்லையெனில் மடக்கு முடிவுப் பெறும்.

10.5.2 for மடக்கு

for மடக்கு ஓர் நுழைவு சோதிப்பு மடக்கு மற்றும் எளிய மடக்காகும். இது கூற்றுகளை மீண்டும் மீண்டும் இயக்கும். for மடக்கு மூன்று

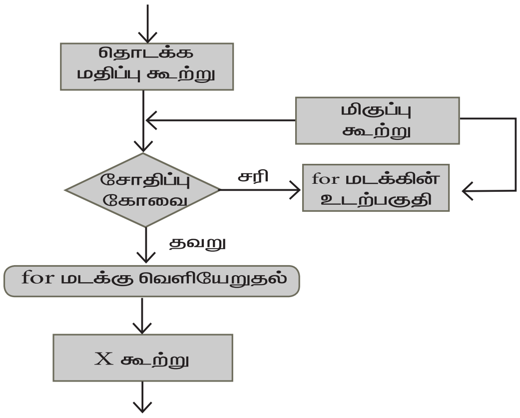
கூற்றுகளை கொண்டிருக்கும் அவை 1. தொடக்க மதிப்பிருத்தல் (Initialization), 2. சோதிப்பு நிபந்தனை அல்லது நிபந்தனை கோவை (Test-expression) மற்றும் 3. மிகுப்பு கோவைகள் (Update expressions). இவை அரைப்புள்ளியால் பிரிக்கப்பட்டிருக்க வேண்டும்.

தொடரியல் :

```
for (தொடக்க மதிப்பிருத்தல்; சோதிப்பு
நிபந்தனை; மிகுப்பு கோவைகள்)
{
        நிரல் கூற்று 1;
        நிரல் கூற்று 2;
        .............
}
நிரல் கூற்று -x;
```

தொடக்க மதிப்பிருத்தல் பகுதி, கட்டுப்பாட்டு மாறியை அறிவிக்க அல்லது மாறியின் மதிப்பிருத்த பயன்படுகிறது. இந்த பகுதி ஒரு முறை மட்டும் இயக்கப்படும். பின்னர் நிரல் பாய்வு, சோதிப்பு நிபந்தனை கோவைக்கு செல்லும். கட்டுப்பாட்டு கோவையின் நிபந்தனை சோதிக்கப்பட்டு, அதன் மதிப்பு தவறு என்று வந்தால் நிரல் கூற்று-x இயங்கும். மதிப்பு சரியென்று வந்தால் for மடக்கின் உடற்பகுதி இயக்கப்பட்டு, பின்னர் நிரல் பாய்வு மிகுப்பு கோவைக்கு அனுப்பி வைக்கப்படும். மிகுப்பு கோவை செயல்பட்ட பின்னர், மீண்டும் சோதிப்பு நிபந்தனை கோவைக்கு அனுப்பி வைக்கப்படும்.

for மடக்கு இயக்கும் முறை மற்றும் பாய்வு படம் 10.6 ல் விளக்கப்பட்டுள்ளது.



படம் 10.6: for மடக்கின் பாய்வு

மேலே கொடுக்கப்பட்டுள்ள நிரலில், முதலில் கட்டுப்பாட்டு மாறி i க்கு தொடக்க மதிப்பிருத்தப்படும், பின்னர் i மாறியின் தொடக்க மதிப்பு 10 என்ற எண்ணுடன் ஒப்பிடப்படும். ஒப்பீட்டின் முடிவு சரி என இருந்தால், நிரல் கட்டுப்பாட்டு பாய்வு, இயக்கப்பட வேண்டிய நிரல் கூற்று cout<< "value of i : " <<i<<endl;

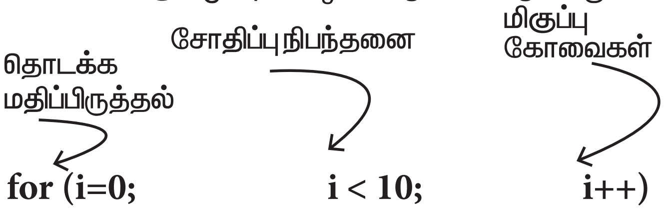
எடுத்துக்காட்டு 10.7: சுழியம் (0) முதல் ஒன்பது (9) வரை உள்ள எண்களை வெளியிட for மடக்கை பயன்படுத்தி நிரல் ஒன்று எழுதுக.

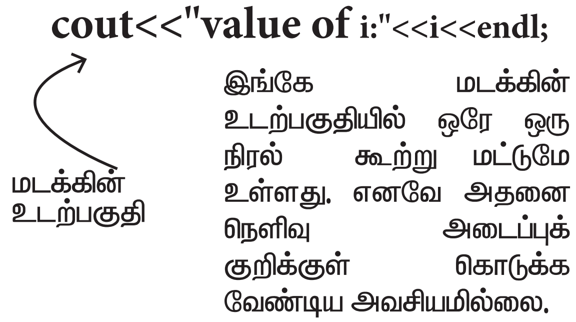
```
#include <iostream>
using namespace std;
int main ()
{
int i;
for(i = 0; i < 10; i ++ )
        cout<< "value of i : " <<i<<endl;
return 0;
}
```

வெளியீடு

```
value of i : 0
value of i : 1
value of i : 2
value of i : 3
value of i : 4
value of i : 5
value of i : 6
value of i : 7
value of i : 8
value of i : 9
```

மேலே கொடுக்கப்பட்டுள்ள நிரலில் for மடக்கு இயங்கும் முறை கீழே கொடுக்கப்பட்டுள்ளது.





என்பதை இயக்கும். பின்னர், நிரல் கட்டுப்பாட்டு பாய்வு for மடக்கின் மிகுப்பு கோவைக்கு செல்லும். அங்கு i ன் மதிப்பு மிகுக்கப்படும். இதைப்போல 0 முதல் 9 வரை மடக்கு இயக்கப்பட்டு எண்கள் வெளியிடப்படும். நிபந்தனை சோதிப்பு தவறாகும்போது, கட்டுப்பாட்டு பாய்வு மடக்கை விட்டு வெளியேறும்.

Chapter 10.indd 165 4/13/2020 6:10:03 PM

எடுத்துக்காட்டு 10.8 for மடக்கை கொண்டு 1 முதல் 10 வரை உள்ள எண்களின் தொடர் கூட்டலை வெளியிடும் நிரல் எழுதுக.

```
#include <iostream>
using namespace std;
int main ()
{
int i,sum=0;
for(i=1; i<=10;i++)
  {
        sum=sum+i;
  }
cout<<"The sum of 1 to 10 is "<<sum;
return 0;
}
```

வெளியீடு

```
        The sum of 1 to 10 is 55
```

for மடக்கின் மாறுபட்ட வடிவங்கள்

for கூற்று C++ மொழியில் ஓர் முக்கிய மடக்கு கூற்றாகும். ஏனெனில், இதனை வேறுபட்ட பல வடிவங்களில் பயன்படுத்தலாம். இவ்வகை வடிவங்கள் நெகிழ்வு மற்றும் பொருந்தும் தன்மையை மடக்கிற்கு வழங்குகிறது. for மடக்கின் பல்வேறு வடிவங்கள் கீழே கொடுக்கப்பட்டுள்ளது.

ஒன்றுக்கும் மேற்பட்ட தொடக்க மதிப்பு மற்றும் மிகுப்புகள்:

for மடக்கில் தொடக்க மதிப்பிருத்த மற்றும் மிகுப்பு கோவையை அறிவிக்க ஒன்றுக்கும் மேற்பட்ட கூற்றுகளை பயன்படுத்தலாம். தொடக்க மதிப்பிருத்தல் மற்றும் மிகுப்பு கோவைகளை ஒன்றுக்கும் மேற்பட்ட கூற்றுகளைக் கொண்டு நிறைவேற்றும் போது அவை காற்புள்ளியில் (,) பிரிக்கப்பட்டிருக்க வேண்டும். எடுத்துக்காட்டு:

```
#include<iostream>
using namespace std;
int main()
{
int i, j
          for(i=0, j=10 ; i<j ; i++,j--)
{
      cout<<"\nThe value of i is"<<i<<"The value of j is "<< j;
}
return 0;
}
```

காள்புள்ளியால் பிரிக்கப்பட்ட, ஒன்றுக்கும் மேற்பட்ட தொடக்க மதிப்பிருத்தல்கள்.

காள்புள்ளியால் பிரிக்கப்பட்ட, ஒன்றுக்கும் மேற்பட்ட மிகுப்பு கோவைகள்.

வெளியீடு

```
The value of i is 0  The value of j is 10
The value of i is 1  The value of j is 9
The value of i is 2  The value of j is 8
The value of i is 3  The value of j is 7
The value of i is 4  The value of j is 6
```

மேலே கொடுக்கப்பட்டுள்ள எடுத்துக்காட்டில் தொடக்க மதிப்பிருத்தலின், i மற்றும் j என்ற இரு வேறு மாறிகள் பயன்படுத்தப்பட்டுள்ளது. அதே போன்று, i++ மற்றும் j++ என்ற இரண்டு மிகுப்புக் கோவைகளை கொண்டுள்ளது. இவ்விரு மாறிகளும் காற்புள்ளியால் பிரிக்கப்பட்டு வரிசை முறையில் வைக்கப்பட்டுள்ளது. அதாவது தொடக்க மதிப்பு இருக்கும் போது i ன் மதிப்பு 0 எனவும் j ன் மதிப்பு 10 எனவும் கருதப்படுகிறது. மிகுப்பு கோவை i++ முதலில் மிகுக்கப்படும், பின்னர் j-- மிகுக்கப்படும்.

பின்னொட்டு மற்றும் முன்னொட்டு செயற்குறிகள்:

பொதுவாக மிகுப்பு கோவை, மிகுப்பு (++) மற்றும் குறைப்பு (--) செயற்குறியை கொண்டிருக்கும். மிகுப்பு கோவை தனியாக பயன்படுத்தப்படும் போது முன்னொட்டு மிகுப்பு அல்லது குறைப்பு செயற்குறியை விட

Chapter 10.indd 166 4/13/2020 6:10:03 PM

பின்னொட்டை அதிகம் பொருத்தமானதாக அமையும். முன்னொட்டு செயற்குறி, பின்னொட்டு செயற்குறியை விட விரைவாக இயக்கப்படுவது இம்முறையை விரும்பக் காரணம் ஆகும்.

விருப்ப கோவைகள் (Optional expressions)

பொதுவாக, **for** மடக்கில் மூன்று பகுதிகள் உள்ளது, அவை தொடக்க மதிப்பு, நிபந்தனை கோவைகள் மற்றும் மிகுப்பு கோவை. இந்த மூன்று கோவைகளும், ஓர் மடக்கில் விருப்ப கோவைகளாக பயன்படுத்தப்படலாம்.

எடுத்துக்காட்டு 10.9 1 முதல் 10 வரையான எண்களின் கூட்டுத்தொகையை காணும் C++ நிரல்

```
#include <iostream>
using namespace std;
int main ()
{
int i, sum=0, n;
cout<<"\n Enter The value of n";
cin>>n;
i =1;
for ( ; i<=n; )
 {
        sum += i;
        ++ i ;
 }
out<<"\n The sum of 1 to " <<n<<"is "<<sum;
return 0;
}
```

வெளியீடு

```
Enter the value of n 5
The sum of 1 to 5 is 15
```

மேலே கொடுக்கப்பட்டுள்ள நிரலில், தொடக்க மதிப்பிருத்தம் மற்றும் மிகுப்பு கோவை தரப்படவில்லை.ஆனால், இரண்டு பகுதிகளையும் குறிக்க அரைப்புள்ளி மட்டும் பயன்படுத்தப்பட்டுள்ளது.

```
for (     ;      i<=n;     )
```

தொடக்க மதிப்பிருத்தம் மற்றும் மிகுப்பு கோவை விடுபட்டுள்ளது

ஒன்று அல்லது இரண்டு கோவைகள் கொடுக்கப்படாதபோது, நிரல் பாய்வுக் கட்டுப்பாடு சோதிப்பு நிபந்தனைக்கு அனுப்பி வைக்கப்படும்.

சூழல் - 3

for மடக்கில் நிபந்தனை சோதிப்பு கோவை கொடுக்கப்படாத போது மடக்கு முடிவற்றதாக இயக்கப்படும்.

நிபந்தனை சோதிப்பு விடுபட்டுள்ளது

```
for( i=0 ;  ;   ++i)
  cout<<"\n Welcome";
```

இந்தக் கூற்று முடிவில்லாமல் தொடர்ந்து இயங்கும்.

அதே போன்று, கீழே கொடுக்கப்பட்டுள்ள மடக்கும், முடிவில்லாமல் தொடர்ந்து இயங்கும்.

அனைத்து மூன்று கோவைகளும் விடுபட்டுள்ளது.

```
for(    ;      ;      )
     cout<<"\n Welcome";
```

இந்தக் கூற்று முடிவில்லாமல் தொடர்ந்து இயங்கும்.

வெற்று மடக்கு (Empty loop)

ஒரு மடக்கில் எந்த கூற்றும் இடம்பெறவில்லை என்றால் அது வெற்று மடக்கு எனப்படும்: பின்வரும் மடக்கு ஒரு வெற்று மடக்கு ஆகும்.

Chapter 10.indd 167 4/13/2020 6:10:03 PM

```
for( i=0 ; i <=5; +=i) ;
```

for மடக்கின் உடற்பகுதி வெற்று கூற்றுகளை கொண்டுள்ளது.

மேலே கொடுக்கப்பட்டுள்ள மடக்கு வெற்றுக் கூற்றைகொண்டுள்ளதால், இது ஒரு காலி மடக்கு எனப்படும்.

```
int i;
for( i=0     ;     i<=5;   ++i) ;
{

    cout<<"\nWe are Indians";
}
```

for மடக்கின் உடற்பகுதி வெற்று கூற்றுகளைக் கொண்டுள்ளது.

மடக்கு அரைப்புள்ளியில் (;) முடிக்கப்பட்டுள்ளது. எனவே மடக்கின் உடற்பகுதியில் உள்ள கூற்றுகள் எதுவும் இயக்கப்படுவதில்லை.

மேலே கொடுக்கப்பட்டுள்ள நிரலில் for மடக்கின் இறுதியில் அரைப்புள்ளி இடம்பெற்றுள்ளதால், நெளிவு அடைப்புக்குறியில் இடம்பெற்றுள்ள மடக்கின் உடற்பகுதி இயக்கப்படாது.

for மடக்கில் மாறி அறிவிப்பு (Declaration of variable in a for loop)

C++ மொழியில் , மாறிகள் for மடக்குக்கு உள்ளும் அறிவிக்கப்படலாம். எடுத்துக்காட்டு,

```
int main ()
{
  int sum = 0;
  for(int i=0;  i<=5;  ++i )
  {
      sum = sum + i;
  }
  cout<<"\nThe variable i cannot access here";
  cout<<"\n The variable sum can access here";
}
```

மாறி (i) for மடக்குக்கு உள்ளே அறிவிக்கப்பட்டுள்ளது

மாறி i மடக்கின் உடற்பகுதியில் மட்டுமே செயல்படுத்தப்படும்.

குறிப்பு

main() உள்ளே எதாவது ஒரு தொகுதிக்குள் அறிவிக்கப்படும் மாறிகள், main() தொகுதிக்குள் எங்கு வேண்டுமென்றாலும் அணுகலாம்.

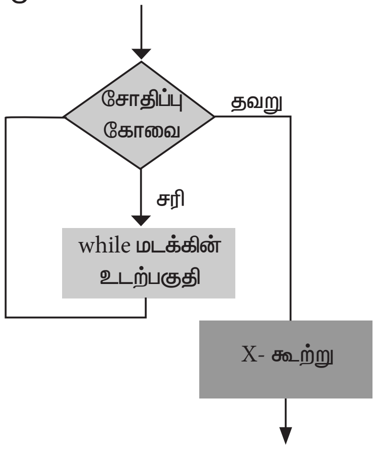
10.5.3 while மடக்கு

while மடக்கும் ஒரு கட்டுப்பாடு பாய்வு கூற்றாகம். இது ஒரு மடக்கினை கொடுக்கப்பட்ட நிபந்தனை சரியாக இருக்கும் வரை, மீண்டும் மீண்டும் இயக்கும். while மடக்கு ஒரு நுழைவு சோதனை மடக்காகும். இதில் சோதிப்பு கோவை முதலில் மதிப்பீடு செய்யப்பட்ட பின்னரே மடக்கினுள் உள்ளே நுழையும்.

while மடக்கின் தொடரியல்:

```
while ( நிபந்தனை சோதிப்புக் கோவை )
{
        மடக்கின் உடற்பகுதி;
}
கூற்று-x;
```

while மடக்கில், நிபந்தனை சோதிப்பு கோவை மதிப்பீடு செய்யப்பட்டு அதன் முடிவு சரி என வந்தால், மடக்கின் உடற்பகுதி இயக்கப்பட்டு மீண்டும் while மடக்கிற்கு கட்டுப்பாடு அனுப்பி வைக்கப்படும்.



பாய்வுப் படம்10.7: while மடக்கின் பாய்வு

நிபந்தனை சோதிப்பு கோவை முடிவு தவறு என வரும்போது, பாய்வுக் கட்டுப்பாடு கூற்று-X அனுப்பி வைக்கப்படும்.

Chapter 10.indd 168 4/13/2020 6:10:04 PM

எடுத்துக்காட்டு 10.10 while மடக்கை பயன்படுத்தி 1 முதல் 10 வரை உள்ள எண்களின் தொடர் கூட்டலை வெளியிடும் நிரல் எழுதுக.

```
#include <iostream>
using namespace std;
int main ()
{
int i=1,sum=0;
while(i<=10)
{
        sum=sum+i;
        i++;
}
cout<<"The sum of 1 to 10 is "<<sum;
return 0;
}
```

வெளியீடு

```
The sum of 1 to 10 is 55
```

மேலே கொடுக்கப்பட்டுள்ள நிரலில், முழு எண் மாறி i யின் மதிப்பு 1 என்றும் sum என்ற மாறியின்மதிப்பு0என்றும்மதிப்பிருத்தப்பட்டுள்ளது. while மடக்கு i < 10 என்ற நிபந்தனையை சோதித்து, அதன் முடிவு சரி எனில் i என்ற மாறியின் மதிப்பு sum மாறியுடன் கூட்டப்படும், பின்னர் i யின் மதிப்பு 1 மிகுக்கப்படும். மீண்டும் நிபந்தனை i < 10 சோதிக்கப்படும். இப்பொழுது 2 < 10, சரி என்பதால் sum என்ற மாறியியுடன் 2 கூட்டப்படும். இது போன்று i யின் மதிப்பு 11 ஆகும் வரை நடைபெறும். இப்பொழுது 11 < 10 என்ற சோதனை தவறு என வருவதால், while மடக்கை விட்டு வெளியே செல்லும். மடக்கை விட்டு வெளியே வந்ததும், sum மாறியின் மதிப்பு வெளியிடப்படும்.

எடுத்துக்காட்டு 10.11 while மடக்கை பயன்படுத்தி 5 எண்களை உள்ளீடாகப் பெற்று அவற்றின் கூட்டுத் தொகை மற்றும் சராசரியை காணும் நிரல் எழுதுக.

```
#include <iostream>
using namespace std;
int main ()
{
int i=1, num, avg, sum=0;
while (i <= 5)
{
        cout<<"Enter the number : ";
        cin>>num;
        sum=sum+num;
        i++;
}
avg=sum/5.0;
cout<<"The sum is "<<sum<<endl;
cout<<"The average is "<<avg;
return 0;
}
```

வெளியீடு

```
Enter the number : 1
Enter the number : 2
Enter the number : 3
Enter the number : 4
Enter the number : 5
The sum is 15
The average is 3
```

மேலே கொடுக்கப்பட்டுள்ள நிரலில், num , avg என்ற மாறி அறிவிக்கப்பட்டு i-ன் மதிப்பு 1 மற்றும் sum - ன் மதிப்பு 0 என தொடங்கி வைக்கப்பட்டுள்ளது. while மடக்கு, i <= 5 என்ற நிபந்தனையை சோதித்து, சரி என முடிவு பெறப்பட்டால், பயனரிடம் இருந்து ஒரு எண் உள்ளீடாகப் பெறப்பட்டு, Sum உடன் கூட்டப்பட்டு பின் i மதிப்பு மிகுக்கப்படும். மீணடும் i <= 5 என்ற நிபந்தனை சோதிக்கப்படுகிறது, மேலும் 2 <=5 என்ற சோதிப்பின் போது இரண்டாவது எண் பயனரிடம் பெறப்பட்டு sum உடன் கூட்டப்படும். இது போன்று i-யின் மதிப்பு 6 வரும் வரை இயக்கப்படும். இந்நிலையில் while மடக்கு மடக்கை விட்டு வெளியேறுவதும் மடக்கை விட்டு வெளியேறியதும் avg கணக்கிடப்பட்டு avg மற்றும் sum வெளியிடப்படும்.

while மடக்கின் மாறுபட்ட வடிவங்கள்

while மடக்கு பல்வேறு வடிவங்களை பெற்றிருக்கலாம். அது வெறுமை மடக்காகவோ அல்லது முடிவற்ற மடக்காகவோ இருக்கலாம். ஒரு வெறுமை while மடக்கின் உடற்பகுதியில் null கூற்றைத் தவிர வேறு எந்த கூற்றும் இடம் பெற்றிருக்காது. Null கூற்று என்பது அரைப்புள்ளி

(;) ஆகும். எடுத்துக்காட்டு

```
int main()
{

     int i=0;
     while(++i < 10000 )

     return 0;

}
```

இந்த மடக்கு எந்த கூற்றையும் பெற்றிருக்கவில்லை. எனவே இது ஒரு காலி மடக்காகும்.

மேலே கொடுக்கப்பட்டுள்ள குறிமுறையில், மடக்கு ஒரு தாமத மடக்காக (delay loop) கொடுக்கப்பட்டுள்ளது. நேர தாமத மடக்கு ஒரு நிரலை சில நொடிகள் இடைநிறுத்தப் பயன்படுகிறது. மடக்கின் உடற்பகுதியில் எந்தஒரு மிகுப்பு கூற்றும் கொடுக்கப்படாத போது while மடக்கு முடிவற்ற மடக்காக செயல் படும்

எடுத்துக்காட்டு

```
int main()
{

     int i = 0;
     while(i < =10)
       cout <<"The value of i is "<<i;
        i++;

      return 0;

}
```

இந்த கூற்று கொடுக்கப்பட்ட மதிப்பை முடிவில்லாமல் தோன்ற செய்யும் ஏனெனில் மடக்கின் உடற்பகுதியில் மிகுப்பு கூற்றில் இடம் பெற்றிருக்கவில்லை

இந்த கூற்று while மடக்கின் பகுதியாக கருதப்படுவதில்லை ஏனெனில் இது நெளிவு அடைப்புக் குறிக்குள் கொடுக்கப்படவில்லை

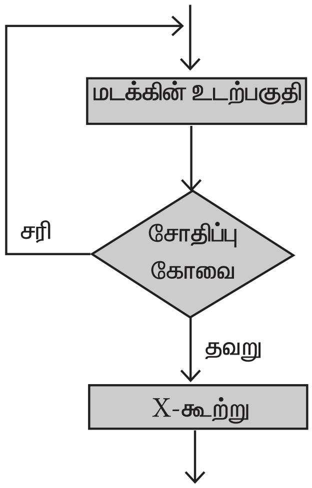
10.5.4 do-while மடக்கு

do-while மடக்கு வெளியேறல் சோதிப்பு மடக்காகும். do while மடக்கினில் மடக்கின் உடற்பகுதி செயல்படுத்தப்பட்ட பின் மடக்கின் இறுதியில் நிபந்தனை சரிபார்க்கப்படும். ஆகையால், நிபந்தனை தவறு என மதிப்பிடப்படும் போதும் மடக்கின் உடற்பகுதி ஒரு முறையேனும் நிறைவேற்றப்படும்.

do-while மடக்கின் கட்டளை அமைப்பு:

```
do
{
        மடக்கின் உடற்பகுதி;
} while(நிபந்தனை சோதிப்பு கோவை);
```

do-while மடக்கின் பாய்வுக் கட்டுப்பாடு மற்றும் பாய்வு படம் கீழே கொடுக்கப்பட்டுள்ளது.



பாய்வுப் படம் 10.8: do-while மடக்கின் பாய்வு கட்டுப்பாடு

Chapter 10.indd 170 4/13/2020 6:10:04 PM

எடுத்துக்காட்டு 10.12 do-while() மடக்கினைப் பயன்படுத்தி 10 முதல் 1 வரை உள்ள எண்களை வெளியிட நிரல் எழுதுக.

```
#include <iostream>
using namespace std;
int main ()
{
int n = 10;
do
{
        cout<<n<<", ";
        n--;
}while (n>0) ;
}
```

வெளியீடு

10, 9, 8, 7, 6, 5, 4, 3, 2, 1,

மேற்கண்ட நிரலில், n என்ற முழு எண் மாறியின் தொடக்க மதிப்பு 10 என இருத்தப்படுகிறது. அடுத்து, n ன் மதிப்பு 10 என வெளியிடப்பட்டு, பின் 1 குறைக்கப்படும். இப்போது 9 > 0 என்ற நிபந்தனை சரிபார்க்கப்பட்டு ,9 என்று வெளியிடப்படும். பின் n என்பது 8 ஆக குறைக்கப்படும். nன் மதிப்பு 0 என ஆகும் வரை இந்த செயல்பாடு தொடரும். நிபந்தனை n > 0 என்பது தவறு என ஆகும் போது, do-while மடக்கு முடிவு பெறும்.

10.5.5 பின்னலான மடக்குகள் (Nesting of loops)

ஒரு மடக்கினுள் இன்னொரு மடக்கு இடம்பெறுமெனில் அது பின்னலான மடக்கு எனப்படும்.

தொடரியல் கீழே தரப்பட்டுள்ளது :'

```
for (தொடக்க மதிப்பு; நிபந்தனைக் கோவை; மிகுப்பு கோவை)
{
      for (தொடக்க மதிப்புதொடக்க மதிப்பு;நிபந்தனைக் கோவை; மிகுப்பு கோவை)
        {
        கூற்று / கூற்றுகள்;
        }
கூற்று / கூற்றுகள்;
}
```

```
while(நிபந்தனை)
{
        while(நிபந்தனை)
        {
        கூற்று / கூற்றுகள்;
        }
கூற்று / கூற்றுகள்;
}
```

```
do
{
கூற்று / கூற்றுகள்;
        do
        {
        கூற்று / கூற்றுகள்;
        }while(நிபந்தனை);
} while(நிபந்தனை);
```

எடுத்துக்காட்டு **10.13** பின்னலான **for** மடக்கினை கொண்டு அணிக்கோவையில் பெருக்கல் வாய்ப்பாட்டினை வெளியிட ஒரு நிரலை எழுது.

```
#include<iostream>
using namespace std;
int main(void)
{
        cout<< "A multiplication table:" <<endl
        <<"  1\t2\t3\t4\t5\t6\t7\t8\t9" <<endl<< "" <<endl;
        for(int c = 1; c < 10; c++)
        {
        cout<< c << "| ";
        for(int i = 1; i< 10; i++)
                {
        cout<<i * c << '\t';
        }
        cout<<endl;
        }
return 0;
}
```

வெளியீடு

```
A multiplication table:
          1   2   3   4   5   6   7   8   9
  1  |    1   2   3   4   5   6   7   8   9
  2  |    2   4   6   8  10  12  14  16  18
  3  |    3   6   9  12  15  18  21  24  27
  4  |    4   8  12  16  20  24  28  32  36
  5  |    5  10  15  20  25  30  35  40  45
  6  |    6  12  18  24  30  36  42  48  54
  7  |    7  14  21  28  35  42  49  56  63
  8  |    8  16  24  32  40  48  56  64  72
  9  |    9  18  27  36  45  54  63  72  81
```

10.6 தாவுதல் கூற்றுகள் (Jump statements)

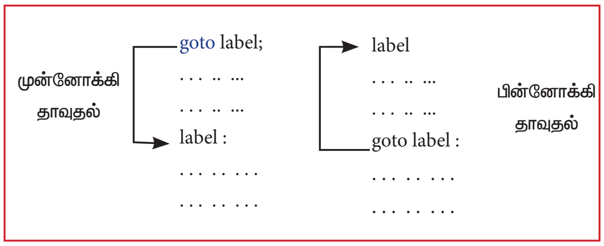
தாவுதல் கூற்று நிரலின் ஓட்டத்தை குறுக்கிட உதவுகிறது. தாவுதல் கூற்றின் வகைகள்

- goto கூற்று
- break கூற்று
- continue கூற்று

10.6.1 goto கூற்று

goto கூற்று நிபந்தனையில்லா கட்டுப்பாட்டு கூற்றாகும். இது நிரலில் ஒரு கூற்றிலிருந்து மற்றொரு கூற்றிற்கு எந்த வித நிபந்தனையும் இன்றி தாவச் செய்யும்.

தொடரியல் 1	தொடரியல் 2
goto label;	**label:**
--------------	-------------
--------------	-------------
--------------	-------------
label:	**goto label;**



மேலே கொடுக்கப்பட்டுள்ள கட்டளை அமைப்பில், முகவரி என்பது குறிப்பெயராகும்- goto label; என்ற கூற்று இயக்கப்படும் போது கட்டுப்பாடு label: என்ற கூற்றினுக்கு தாவும் பின் அதற்கு கீழ் வரும் இடம் பெற்றுள்ள கூற்றுகள் இயக்கப்படும்.

எடுத்துக்காட்டு : 10.14 goto கூற்றை பயன்படுத்தி கொடுக்கப்பட்ட எண்களின் முதல் ஐந்து ஒற்றைப்படை எண்களைக் காண்பிக்கும் நிரல்

```
# include <iostream>
using namespace std;
int main()
{
int n=1;
jump:
{
  if(n<10)
  { // Control of the program move to jump:
        cout<<n<<'\t';
        n+=2;
        goto jump;
  }
  else
        return 0;
} }
```

வெளியீடு

```
1    2    5    7    9
```

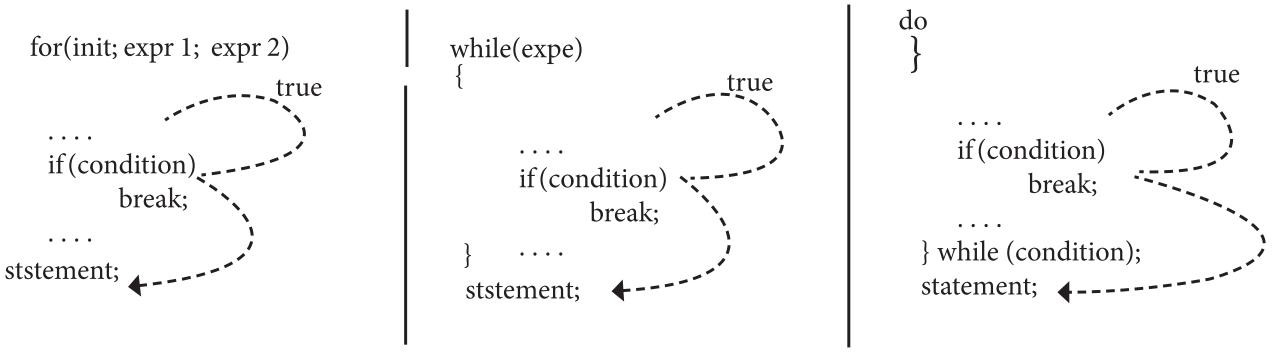
மேலேயுள்ள நிரலில் முதல் ஐந்து ஒற்றைப்படை எண்கள் காண்பிக்கப்படுகின்றன. n என்பது 10 க்கும் குறைவானதாக இருந்தால், அறிக்கையை jump கூற்றிற்கு மாற்றுகிறது. n 10 ஐ விட அதிகமாக இருந்தால் கட்டுப்பாடு வளையிலிருந்து வெளியேறும்.

Chapter 10.indd 172 4/13/2020 6:10:04 PM

10.6.2 break கூற்று

break கூற்று ஒரு தவுதல் கூற்று. இது மடக்கின் இயக்கத்தை நிறுத்தி கட்டுபாட்டை மடக்கின் உடற்பகுதிக்கு வெளியே எடுத்துச் சென்று மற்ற கூற்றுகளை இயக்குகிறது

கீழே கொடுக்கப்பட்டுள்ள படம் ஓர் மடக்கில் break கூற்று இயங்கும் முறையை காட்டுகிறது.



படம் 10.14: break கூற்று for, while மற்றும் do-while loop

எடுத்துக்காட்டு 10.15 break கூற்றை பயன்படுத்தி எண்ணிக்கையை காணும் நிரல்

```
#include <iostream>
using namespace std;
int main ()
{
int count = 1;
do
{
cout<< "Count : " << count <<endl;
if( count > 3)
 {
        break;
 }
count ++;
}while( count < 20 );
return 0;
}
```

வெளியீடு

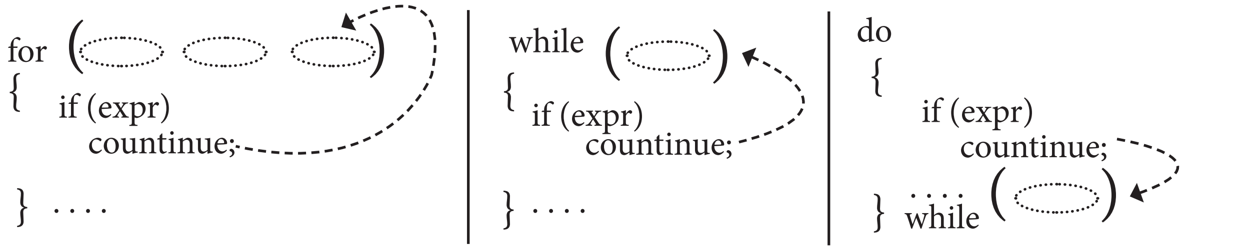
```
Count : 1
Count : 2
Count : 3
```

மேலே கொடுக்கப்பட்டுள்ள எடுத்துக்காட்டில், while மடக்கு 20 முறை இயக்கப்படும் ஆனால் countன் மதிப்பு 5 ஆகிய பின் break கூற்று மடக்கை முடித்து வைக்கும்.

10.6.3 continue கூற்று

Continue கூற்று break கூற்றைப்போல் செயல்படுகிறது. ஆனால், மடக்கை முடித்து வைப்பதற்கு பதிலாக, மடக்கை அடுத்து சுழற்சிக்கு இட்டுச்செல்லும். ஓர் மடக்கினுள் Continue கூற்று இயக்கப்படும் பொழுது, continue கூற்றை தொடர்ந்து வரும் கூற்றுகள் தவிர்க்கப்பட்டு, மடக்கின் அடுத்த சுழற்சியை தொடங்க செய்யும்.

கீழே கொடுக்கப்பட்டுள்ள படம் continue கூற்று இயங்கும் முறையை விளக்குகிறது.



continue கூற்றின் பாய்வு

எடுத்துக்காட்டு 10.16 continue கூற்றை பயன்படுத்தி 6 - ஐ தவிர ஒன்று முதல் பத்து வரையான எண்களை அச்சிடும் C++ நிரல் எழுதுக.

```
#include <iostream>
using namespace std;
 int main()
{
for (int i = 1; i<= 10; i++)
 {
if (i == 6)
continue;
else
cout<<i<< " ";
}
return 0;
}
```

வெளியீடு

```
1 2 3 4 5 7 8 9 10
```

மேலே கொடுக்கப்பட்டுள்ள எடுத்துக்காட்டில் for மடக்கு 10 முறை இயக்கப்படும், ஆனால் i மாறி 6 ஆகும் போது continue கூற்றால் கட்டுப்பாடு for மடக்கிற்கு அனுப்பி வைக்கப்படும்.

break மற்றும் continue ஒப்பீடு

Break	Continue
break மடக்கினை விட்டு வெளியேற செய்யும்.	continue கூற்று மடக்கை விட்டு வெளியேற செய்யாது.
இது சுழற்சியை முடித்து வைக்கும்.	இது சுழற்சியை தவிர்க்கச் செய்யும்.
இந்த கூற்றை இயக்கப்படும் பொழுது கட்டுப்பாட்டை மடக்கினை விட்டு வெளியேறி மடக்கிற்கு அடுத்து வரும் கூற்றுகளை இயக்கும்.	இந்த கூற்று இயக்கப்படும் பொழுது மடக்கைவிட்டு வெளியேறாமல் மடக்கின் அடுத்த சுழற்சிக்கு தாவச் செய்யும்.
break கூற்று மடக்கு மற்றும் switch கூற்றுடன் பயன்படுத்தலாம்.	continue கூற்று மடக்கில் மட்டுமே பயன்படுத்தப்படும், switch கூற்றுடன் பயன்படுத்தப்படாது.

செய்முறைப் பயிற்சி

கீழ்காணும் சிக்கல்களை தீர்க்கும் C++ நிரல்களை எழுதுக.

1. C++ குறியுருத் தொகுதியிலுள்ள ஏதேனும் ஒரு குறியுருவை உள்ளீடாகப் பெற வேண்டும். பெறப்பட்ட குறியுரு எழுத்து, எண் அல்லது சிறப்புக் குறியுரு இவற்றில் எது என்பதைக் காட்டும் C++ நிரல் ஒன்றை எழுதுக.
2. கொடுக்கப்படும் குறியுரு ஆங்கில பெரிய எழுத்தா, சிறிய எழுத்தா, எண்ணா அல்லது வேறு ஏதேனும் குறியுருவா என்பதை ASCII மதிப்புகளைக் கொண்டு கண்டறியும் C++ நிரல் ஒன்றை எழுதுக. ASCII மதிப்புகள் கீழே கொடுக்கப்பட்டுள்ளது.

குறியுருக்கள்	**ASCII** மதிப்புகள்
'0' - '9'	48 - 57
'A' - 'Z'	65 - 90
'a' - 'z'	97 - 122
மற்ற குறியுருக்கள்	0- 255 மேலே குறிப்பிடப்பட்டுள்ள மதிப்புகள் நீங்கலாக

3. ஒரு முழு எண்ணின் தொடர் பெருக்கலை (Factorial) கணக்கிடும் C++ நிரல் ஒன்றை எழுதுக.
4. ஃபிபோனாசி எண் வரிசை (0 1 1 2 3 5 8 …….) கண்டறியும் C++ நிரல் ஒன்றை எழுதுக.
5. பின்னலான மடக்குகளைப் பயன்படுத்தி, பின்வரும் வடிவங்களை அச்சிடுவதற்கான C++ நிரல் ஒன்றை எழுதுக.

(அ)	(ஆ)
A	5 4 3 2 1
A B	5 4 3 2
A B C	5 4 3
A B C D	5 4
A B C D E	5
A B C D E F	

நினைவில் கொள்க

- கணிப்பொறி நிரல் என்பது ஒரு குறிப்பிட்ட பணியை நிறைவேற்ற கொடுக்கப்படும் கூற்றுகள் அல்லது கட்டளைகளின் தொகுப்பாகும்.
- C++ நிரலில் இரு வகையான கூற்றுகள் உள்ளன. (1) வெற்றிட கூற்றுகள் (2) கலவை கூற்றுகள்
- கட்டுப்பாடு, நிரலின் ஒரு பகுதியிலிருந்து இன்னொரு பகுதிக்குத் தாவுவதற்குக் காரணமாக நிரல் கூற்றுகள் கட்டுப்பாட்டு கட்டமைப்புகள் எனப்படுகின்றன.
- C++ ல் மூன்று வகையான கட்டளை கூற்றுகள் உள்ளன: (1) வரிசை முறை கூற்றுகள் (Sequence Statement) (2) தேர்ந்தெடுப்பு கூற்றுகள் (Selection Statement) (3) சுழற்சி முறை (Iteration Statement)
- if மற்றும் switch கூற்றுகள் தேர்வு கூற்றுகள் ஆகும்.
- if..else கூற்றின் மாற்று வழிமுறை நிபந்தனை செயற்குறி ஆகும் Switch கூற்று ஒரு பல வழி கிளைப் பிரிப்பு கூற்றாகும்.
- மடக்குகள், ஒரு கட்டளைத் தொகுதியை, நிபந்தனையின் அடிப்படையில் குறிப்பிட்ட தடவைகள் திரும்ப திரும்ப நிறைவேற்றுகின்றன.
- C++ மொழியில் மூன்று வகையான சுழற்சி செயல் அல்லது மடக்குகள் உள்ளன. For, while,do-while
- C++ தாவும் (Jump) கூற்றுகள் goto, break,continue ஆகும்.

Chapter 10.indd 175 4/13/2020 6:10:05 PM

மதிப்பீடு

பகுதி-அ

சரியான விடையை தேர்வு செய்யவும் :

1. வெற்றுக்கூற்றின் மாற்றுப் பெயர் என்ன?
   (அ) கூற்று அல்லா (ஆ) காலிக் கூற்று (இ) void கூற்று (ஈ) சுழியக் கூற்று
2. C++ல் குறிமுறைத் தொகுதிகள் இந்தக் நிறுத்தற்குறிக்குள் கொடுக்கப்பட வேண்டும்:
   (அ) { } (ஆ) [ ] (இ) ( ) (ஈ) < >
3. சுழற்சியில், மீண்டும் மீண்டும் இயக்கப்படும் குறிமுறைத் தொகுதிகள் இவ்வாறு அழைக்கப்படுகிறது:
   (அ) நிபந்தனை (ஆ) மடக்கு (இ) கூற்று (ஈ) மடக்கின் உடற்பகுதி
4. பல வழி கிளைப் பிரிப்புக் கூற்று:
   (அ) if (ஆ) if … else (இ) switch (ஈ) for
5. சுழற்சிக் கூற்றுகள் எத்தனை வகைப்படும்?
   (அ) 2 (ஆ) 3 (இ) 4 (ஈ) 5
6. for (int i=0; i<10; i++) என்ற மடக்கு எத்தனை முறை இயங்கும்?
   (அ) 0 (ஆ) 10 (இ) 9 (ஈ) 11
7. பின்வருவனவற்றில் எது வெளியேறல் சோதிப்பு மடக்கு?
   (அ) for (ஆ) while (இ) do…while (ஈ) if…else
8. தாவுதல் கூற்றுகளின் சிறப்புச் சொற்களில் பொருந்தா ஒன்றை கண்டுபிடி.
   (அ) break (ஆ) switch (இ) goto (ஈ) continue
9. பின்வருவனவற்றில் எது நுழைவு சோதிப்பு மடக்கு?
   (அ) for (ஆ) while (இ) do…while (ஈ) if…else
10. ஒரு மடக்கு அதன் உடற்பகுதியில் மற்றொரு மடக்கை பெற்றிருப்பது:
    (ஆ) பின்னலான மடக்கு (ஆ) உள் மடக்கு
    (இ) உள்ளிணைந்த மடக்கு (ஈ) மடக்குகளின் பின்னல்

பகுதி - ஆ

குறு வினாக்கள் (2 மதிப்பெண்கள்) :

1. வெற்றுக்கூற்று மற்றும் கூட்டுக்கூற்று என்றால் என்ன?
2. தேர்ந்தெடுப்புக் கூற்றுகள் என்றால் என்ன? அதன் வகைகளை எழுதுக.
3. பின்வரும் நிரலில் உள்ள பிழைகளை சரிசெய்க:

```
if (x=1)
  p= 100;
else
  p = 10;
```

4. பின்வரும் நிரலின் வெளியீடு என்ன?

```
int year;
cin >> year;
  if (year % 100 == 0)
          if ( year % 400 == 0)
                  cout << "Leap";
  else
```


Chapter 10.indd 176 4/13/2020 6:10:05 PM

cout << "Not Leap year";

If the input given is (i) 2000 (ii) 2003 (iii) 2010?

5. பின்வரும் நிரலில் உள்ள பிழைகளை சரிசெய்க:

```
for (int i=2; i<=10 ; i+=2)
  cout << i;
```

6. 21 முதல் 30 வரை தொடர்ச்சியாக எண்களை அச்சிடுவதற்கான for மடக்கை எழுதுக.
7. 2, 4, 6, 8 .... 20 என்ற தொடர் வரிசையை அச்சிடுவதற்கான while மடக்கை எழுதுக.
8. if கூற்றுடன், ?: மும்ம செயற்குறியை ஒப்பிடுக.

பகுதி- இ

சிறு வினாக்கள் (3 மதிப்பெண்கள்) :

1. பின்வரும் if - else கூற்றுக்கு நிகரான நிபந்தனை கூற்றாக மாற்றுக:

```
if (x >= 10)
        a = m + 5;
else
        a = m;
```

2. பின்வரும் நிரல் கூற்றுகள் சரியாக இயங்கும் வகையில் அவற்றை மாற்றி எழுதுக.

```
v = 5;
do;
{
  total += v;
  cout << total;
while v <= 10
```

3. கொடுக்கப்பட்ட எண்ணின் பெருக்கல் வாய்ப்பாட்டை அச்சிடும் C++ நிரல் ஒன்றை எழுதுக.
4. switch கூற்றின் கட்டளை தொடரை எழுதி அதன் பயன்களை பட்டியலிடுக.
5. பின்வரும் எண் தொடரை அச்சிடுவதற்கான நிரல்களை எழுதுக.

   (a) 1 4 7 10...... 40

பகுதி – ஈ

பெரு வினாக்கள் (5 மதிப்பெண்கள்) :

1. கட்டுப்பாட்டு கூற்றுகளை பொருத்தமான எடுத்துக்காட்டுடன் விளக்குக.
2. நுழைவு சோதிப்பு மடக்கு என்றால் என்ன? ஏதேனும் ஒரு நுழைவு சோதிப்பு மடக்கை பொருத்தமான எடுத்துக்காட்டுடன் விளக்குக.
3. LCM மற்றும் GCD போன்றவற்றை கணக்கிடுவதற்கான நிரல்களை எழுதுக.
4. கீழ்காணும் எண் தொடர்களை கணக்கிடுவதற்கான நிரல்களை எழுதுக

   (a) $x- \frac{x^2}{2!} + \frac{x^3}{3!} - \frac{x^4}{4!} + \frac{x^5}{5!} - \frac{x^6}{6!}$

   (b) $x+\frac{x^2}{2} + \frac{x^3}{3} +....+ \frac{x^n}{n}$

5. கொடுக்கப்பட்ட எண் தொடரின் கூட்டுத் தொகையை கணக்கிடும் நிரல் ஒன்றை எழுதுக.

   $S = 1 + x + x^2 +..... + x^n$

துணை நூல்கள்:

(1) Object Oriented Programming with C++ (4th Edition), Dr. E. Balagurusamy, Mc.Graw Hills.
(2) The Complete Reference C++ (Forth Edition), Herbert Schildt. Mc.Graw Hills.
(3) Computer Science with C++ (A text book of CBSE XI and XII), Sumita Arora, Dhanpat Rai & Co.