அலகு **I**

## பாடம் 3

# வரையெல்லை

ZDBQDJ

### கற்றலின் நோக்கங்கள்

இந்தப் பாடப்பகுதியைக் கற்றபின் மாணவர்கள் அறிந்துக் கொள்வது,

- வரையெல்லையை பற்றி புரிந்து கொள்ளல்
- LEGB விதியை நடைமுறைப்படுத்துதல்
- தொகுதிகள் (Modules) பற்றி புரிந்து கொள்ளல்
- நிரலாக்க மொழியில் அணுகுதலின் கட்டுப்பாட்டை செயலாக்கம் பற்றிஅறிதல்.

## 3.1 அறிமுகம்

வரையெல்லை என்பது மாறிகள், அளபுருக்கள் மற்றும் செயற்கூறுகளின் அணுகியல்பை நிரலின் ஒரு பகுதியில் இருந்து மற்றொரு பகுதிக்கு குறிப்பதாகும். அதாவது, நிரலின் எந்தப் பகுதியை அணுக அல்லது பயன்படுத்த முடியும் என்பதைக் குறிக்கிறது. பொதுவாக, நிரலில் வரையறுக்கப்பட்ட ஒவ்வொரு மாறியும் முழுதளாவிய வரையெல்லையைக் கொண்டுள்ளன. ஒரு முறை வரையறுக்கப்பட்டால், நிரலின் ஒவ்வாரு பகுதியும் அந்த மாறியை அணுக முடியும். ஆனால், ஒரே ஒரு வரையறைக்குள் மாறிகளின் வரையெல்லை உட்படுத்துவது சிறந்த வழிமுறை ஆகும். இதில் எதிர்பாராத விதமாக செயற்கூறுக்கு உள்ளே உள்ள மாறிகளில் ஏற்படும் மாற்றங்கள் செயற்கூறுவுக்கு வெளியே எந்த மாற்றத்தையும் ஏற்படுத்தாது.

## 3.2 மாறியின் வரையெல்லை

நிரலாக்க மொழியில் உள்ள மாறிகளின் வரையெல்லையை அறிந்து கொள்வதற்கு மாறிளைப் பற்றி தெரிந்து கொள்ளுதல் அவசியம். முக்கியமாக, இவைகள் நினைவகத்தில் உள்ள ஒரு பொருளின் முகவரியாகும். (குறிப்புகள் (அ) சுட்டுகள்) ஒரு பொருளுக்கு ஒரு மாறியை : = குறியுடன் இருத்தும்போது மாறியும் பொருளும் ஒன்றாகப் பிணைக்கப்படுகிறது. ஒன்றிக்கு மேற்பட்ட மாறிகளுக்கு மதிப்புகள் இருக்க முடியும் (Map).

**குறிப்பு**

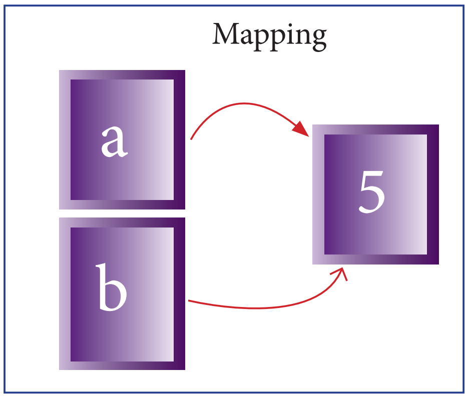
மாறியின் பெயரை ஒரு பொருளுடன் பிணைக்கும் செயல்முறையே மேப்பிங் (Mapping) எனப்படும். : = என்ற குறியீடு செயல்குறி நிரலாக்க மொழியில் மாறி மற்றும் பொருளை = (சமக்குறி) (Map) இருக்கிறது.

நிரலாக்க மொழி இதைப்போன்ற அனைத்து மேப்பிங்களையும் namespaces உடன் கண்காணிக்கிறது. namespaces என்பது மாறியின் பெயரை பொருளுடன் மேப்பிங் செய்வதற்கான கொள்கலனாகும். இவைகளை Dictionary, சொற்களைக் கொண்ட List மற்றும் அதற்கான அர்த்தங்களாகவும் எடுத்துக் கொள்ளலாம். Dictionary-ல் வார்த்தைகள் அதன் அர்த்தங்களுடனும், பெயர்கள் பொருள்களுடனும் (name:=object) மேப் செய்யப்பட்டுள்ளன. இது நீங்கள் தேர்வு செய்த பெயரால் பொருளை அணுக அனுமதிக்கிறது. பின்வரும் எடுத்துக்காட்டில் a என்பது 5 என்ற முழு எண்ணுடன் முதலில் மேப் செய்யப்படுகிறது. இங்கு a என்பது மாறியின் பெயர். முழு எண் மதிப்பு 5 என்பது பொருளாகும். பின்னர், a என்பது b-ல் இருத்தப்படுகிறது. அதாவது, b என்பது a-ல் உள்ள அதே முழு எண் மதிப்பாகிறது.

மேப்பிங்

1. a:=5

2. b:=a

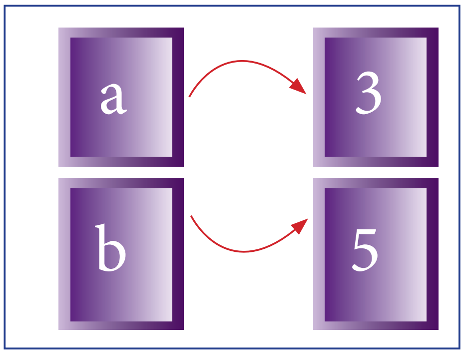


மேலே கொடுக்கப்பட்டுள்ள கூற்றுகளில் சில மாற்றத்தை கீழே வருமாறு காண்போம்.

1. a:=5
2. b:=a
3. a:=3

அதன் பிறகு a-ன் மதிப்பை 3 என மாற்றினால் தொடக்க நிரலர் b-ன் மதிப்பும் 3 என மாறும் என்று கருதுவார். ஆனால், b-ன் மதிப்பு தற்பொழுதும் 5 என்ற முழு எண் மதிப்பாகவே இருக்கும். a-ன் மதிப்பு மட்டுமே 3 என்ற முழு எண் மதிப்பாக மாற்றம் பெற்றிருக்கும்.

a-வின் மதிப்பை மாற்றிய பிறகு மேப்பிங்



மாறியின் வரையெல்லை என்பது அது குறிமுறையில் எங்கு புலப்படுகிறதோ அல்லது காணப்படுகிறதோ அந்தப் பகுதியாகும். அதை அணுகுவதற்கு எந்த முன்னொட்டையும் பயன்படுத்த வேண்டிய அவசியமில்லை. ஒரு எடுத்துக்காட்டு,

```
1. Disp():
2.       a:=7
```

a-வின் மதிப்பை குறிமுறைக்கு வெளியே காண்பிக்க முற்படும்போது நிரல் **“name ‘a’ is not defined”** என்ற பிழைச் செய்தியை வெளியிடும். ஏனெனில், மாறியின் வாழ்நாள் குறிமுறையின் இறுதிவரை மட்டுமே ஆகும். **ஒரு மாறி குறிமுறையில் பயன்படும் நேரமே அதனுடைய வாழ்நாள் 'Life time'** எனப்படும்.

## 3.3 LEGB விதிமுறை

வரையெல்லை என்பது சரியான மதிப்பை பெறுவதற்காக மாறிகள் எந்த வரிசையில் பொருளுடன் Map செய்யப்பட வேண்டும் என்பதை வரையறுக்கிறது. கீழே கொடுக்கப்பட்ட எளிய எடுத்துக்காட்டைக் காண்போம்.

```
1.  x:= 'outer x variable'
2. display():
3.    x:= 'inner x variable'
4.    print x
5.  display()
```

மேலே உள்ள கூற்றுகளை இயக்கும்போது கூற்று (4) மற்றும் (5) பின்வரும் விடையைக் காண்பிக்கிறது.

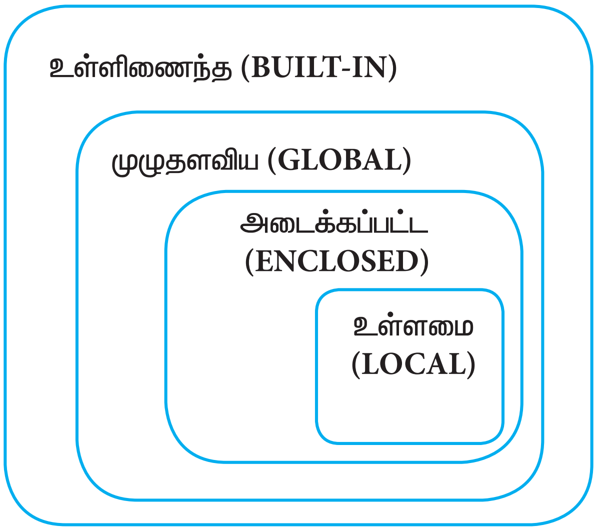
**வெளியீடு**

```
outer x variable
inner x variable
```

மேலே உள்ள கூற்றுகள் வெவ்வேறு வெளியீடுகளைத் தருகிறது. ஏனெனில், மாறி *x* என்பது வெவ்வேறு வரையெல்லைகளில் உள்ளது. ஒன்று display ( ) என்ற செயற்கூறுவுக்கு உள்ளேயும், மற்றொன்று அதன் மேல் கூற்றிலும் உள்ளது. **‘Outer *x* Variable’** என்ற மதிப்பு, *x* என்பது செயற்கூறுவின் வரையறைக்கு வெளியில் அணுகப்படும்போது வெளியிடப்படுகிறது. ஆனால் display ( ) செயற்கூறு இயக்கப்படும்போது, **“inner *x* Variable”** என்ற மதிப்பு அச்சிடப்படுகிறது. இது செயற்கூறு வரையறைக்குள் உள்ள *x*-ன் மதிப்பாகும். மேலே கொடுக்கப்பட்டுள்ள எடுத்துக்காட்டில், ஒரு மாறி எந்த வரையெல்லையில் எடுத்துக் கொள்ளப்பட வேண்டும் என்பதை தீர்மானிக்க ஒரு விதிமுறை பின்பற்றப்படுவதை கணித்துக் கொள்ள முடிகிறது.

LEGB விதி வரையெல்லை தேடப்பட வேண்டிய (Scope resolution) வரிசையை தீர்மானிக்கப் பயன்படுகிறது. வரையெல்லைகள் பின்வருமாறு படிநிலை முறையில் பட்டியலிடப்பட்டுள்ளன. (பெரியதிலிருந்து சிறியது)

| | |
|---|---|
| உள்ளமை Local(**L**) | செயற்கூறு / இனக்குழுவிற்கு உள்ளே வரையறுக்கப்பட்டவை |
| இணைக்கப்பட்ட Enclosed(**E**) | பின்னலான செயற்கூறுகளுக்குள் வரையறுக்கப்பட்டவை |
| முழுதளவிய Global(**G**) | மேல்நிலையில் வரையறுக்கப்பட்டவை |
| உள்ளிணைந்த Built-in (**B**) | உள்ளிணைந்த செயற்கூறுகளில் (கூறுகள்) உள்ள முன்னரே வரையறுக்கப்பட்ட பெயர்களாகும். |

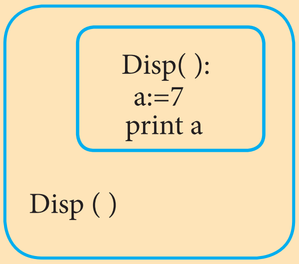


## 3.4 மாறியின் வரையெல்லை வகைகள்

4 வகையான வரையெல்லைகள் உள்ளன. அவற்றை ஒன்றன்பின் ஒன்றாகக் காண்போம்.

### 3.4.1 உள்ளமை வரையெல்லை (Local Scope)

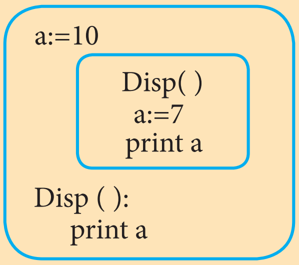
உள்ளமை வரையெல்லை, நடப்பு செயற்கூறில் வரையறுக்கப்பட்ட மாறிகளைக் குறிக்கும். செயற்கூறு, எப்பொழுதும் மாறியின் பெயரை முதலில் அதன் உள்ளமை வரையெல்லையில் பார்வையிடும் அந்த வரையெல்லையில் இல்லையென்றால் மட்டுமே வெளி வரையெல்லையில் சோதிக்கும். இந்த எடுத்துக்காட்டை காண்போம்.

| 1. **Disp():**<br>2. a:=7<br>3. print a<br>4. Disp() | முழுநிரல் | நிரலின் வெளியீடு<br>7 |
|---|---|---|
| |  | |

மேலே உள்ள குறிமுறையை இயக்கும்போது, மாறி a என்பது 7 என்ற மதிப்பை வெளியிடுகிறது. ஏனெனில், இது உள்ளமை வரையெல்லையில் வரையறுக்கப்பட்டு, அங்கேயே அச்சிடப்படுகிறது.

### 3.4.2 முழுதளாவிய வரையெல்லை

நிரலின் அனைத்து செயற்கூறுகளுக்கும் வெளியே அறிவிக்கப்பட்ட மாறிகள் முழுதளாவிய மாறிகள் எனப்படும். அதாவது, முழுதளாவிய மாறிகளை நிரலின் அனைத்து செயற்கூறுகளுக்கு உட்புறமும், வெளிப்புறமும் அணுக முடியும். பின்வரும் எடுத்துக்காட்டைக் காண்போம்.

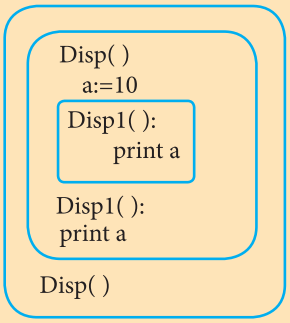
| | முழுநிரல் | நிரலின் வெளியீடு |
|---|---|---|
| 1. a:=10<br>2. **Disp()**:<br>3. a:=7<br>4. print a<br>5. Disp()<br>6. print a |  | 7<br>10 |

மேலே உள்ள குறிமுறையில் disp ( ) என்ற செயற்கூறு அழைக்கப்படும்போது, அதனுள் வரையறுக்கப்பட்டிருக்கும் மாறி **a**, 7 என்ற மதிப்பை வெளியிடும், பின்னர் இது 10 என்ற மதிப்பை வெளியிடும் ஏனெனில் **a** என்ற மாறி முழுதளாவிய வரையெல்லையில் வரையறுக்கப்பட்டுள்ளது.

### 3.4.3. அடைக்கப்பட்ட வரையெல்லை

அனைத்து நிரலாக்க மொழிகளும் பின்னலான செயற்கூறுகளைக் கொடுக்க அனுமதிக்கின்றன. ஒரு செயற்கூறின் (வழிமுறை) உள்ளே மற்றொரு செயற்கூறு அடைக்கப்பட்டிருந்தால் அது **பின்னலான செயற்கூறு** எனப்படும். **மற்றொரு செயற்கூறு வரையறையை, தன்னுள் கொண்ட ஒரு வெளி செயற்கூறினுள் ஒரு மாறி அறிவிக்கப்பட்டால், உள்செயற்கூறானது, வெளி செயற்கூறினுள் உள்ள மாறிகளை அணுக முடியும். இதுவே, அடைக்கப்பட்ட வரையெல்லை எனப்படும்.**

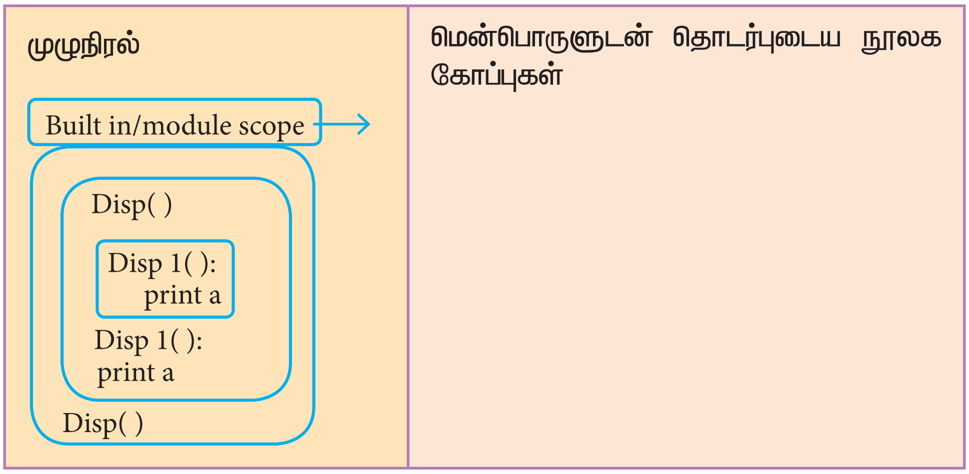
நிரல்பெயர்ப்பி அல்லது தொகுப்பான் ஒரு நிரலில் மாறியை தேடும்பொழுது அது முதலில் உள்ளமை வரையெல்லையில் தேடும். பின்னர் அடைக்கப்பட்ட வரையெல்லையில் தேடும். பின்வரும் எடுத்துக்காட்டை காண்போம்.

| | முழுநிரல் | நிரலின் வெளியீடு |
|---|---|---|
| 1. Disp():<br>2. a:=10<br>3. Disp1():<br>4. print a<br>5. Disp1()<br>6. print a<br>7. Disp() |  | 10<br>10 |

மேலே குறிப்பிட்ட எடுத்துக்காட்டில் Disp1 ( ) என்ற செயற்கூறு Disp ( ) என்ற செயற்கூறினுள் வரையறுக்கப்பட்டுள்ளது. ‘a’ என்ற மாறி Disp ( ) என்ற செயற்கூறினுள் வரையறுக்கப்பட்டிருந்தாலும் Disp 1 ( ) என்ற செயற்கூறும் அதைப் பயன்படுத்த முடியும். ஏனெனில் இந்த செயற்கூறு Disp ( ) என்பதன் உறுப்பாகும்.

### 3.4.4. உள்ளிணைந்த வரையெல்லை

இறுதியாக, நாம் விரிந்த வரையெல்லை பற்றி விவாதிப்போம். நிரல்பெயர்ப்பி அல்லது தொகுப்பாணை தொடங்கும் பொழுது உள்ளிணைந்த வரையெல்லையனானது நிரல் வரையெல்லையில் ஏற்கனவே கொடுக்கப்பட்ட அனைத்து பெயர்களையும் கொண்டிருக்கும். நிரலாக்க மொழியின் நூலக செயற்கூறினுள் வரையறுக்கப்பட்ட மாறி அல்லது தொகுதி உள்ளிணைந்த வரையெல்லையைக் கொண்டுள்ளது. இவைகள், நூலக கோப்புகள் நிரலில் செயல்பட தொடங்கியவுடன் இறக்கப்படும்.



பொதுவாக செயற்கூறுகள் அல்லது தொகுதி மட்டுமே மென்பொருளுடன் தொகுப்பாக வருகிறது. எனவே, இவைகள் உள்ளிணைந்த வரையெல்லையின் கீழ் வருகின்றன.

## 3.5 தொகுதி (Module)

நிரலின் ஒரு பகுதியே தொகுதியாகும். நிரல்கள் ஒன்று அல்லது அதற்கு மேற்பட்ட தனித்து உருவாக்கப்பட்ட தொகுதிகளால் அமைக்கபடுகிறது. ஒரு தொகுதி தொடர்புடைய பல கூற்றுகளை கொண்டுள்ளது. தனித்த நிலையில் தொகுதிகள் சரியாக வேலை செய்கிறது மற்றும் பிற தொகுதிகளோடு ஒருங்கிணைக்கப் படுகிறது. ஒரு நிரலை எளிதாக்கவும், பிழை திருத்தவும் அது தொகுதிகளாக பிரிக்கப்பட்டு வெளியீடு பெறப்படுகிறது. ஒரு கணிப்பொறி நிரலை பல துணை நிரல்களாக பிரிக்கும் செயல்முறையே தொகுதி நிரலாக்கம் எனப்படும். தொகுதி நிரலாக்கத்திற்கு எடுத்துக்காட்டுகள், செய்லமுறைகள், துணை நிரல்கள் மற்றும் செயற்கூறுகள்.

### 3.5.1 தொகுதியின் பண்புயியல்புகள்

தொகுதி பண்பியல்கள் பின்வருமாறு:

1. தொகுதிகள் தரவு, தகவல் மற்றும் தருக்க செயலாக்கத்தைக் கொண்டுள்ளன.
2. தொகுதிகள் தனியாக தொகுக்கப்பட்டு நூலகத்தில் சேமிக்கப்படும்.
3. தொகுதிகள் நிரலில் சேர்க்க முடியும்.
4. ஒரு பெயரையும், சில அளபுருக்களையும் பயன்படுத்தி தொகுதி பிரிவுகள் செயல்படுத்தப்படுகின்றன.
5. ஒரு தொகுதியின் பிரிவுகள் மற்ற தொகுதிகளால் பயன்படுத்தப்படுகின்றன.

### 3.5.2 தொகுதி நிரலாக்கத்தின் பயன்கள்

- குறைந்த வரிகளைக் கொண்ட குறிமுறையை எழுதினால் போதுமானது.
- மறுபயனாக்கத்திற்கும் பலமுறை குறிமுறை தட்டச்சு செய்வதை தவிர்ப்பதற்கு, ஒற்றை செயல்முறையை உருவாக்க வேண்டும்.
- நிரல்கள் மிக எளிதாக வடிவமைக்கப் படுகின்றன. ஏனெனில், முழு குறிமுறையும்

சிறிய பகுதிகளாக பிரிக்கப்பட்டு சிறிய குழுவினரால் கையாளப்படுகிறது.

- தொகுதி நிரலாக்கம் பல நிரலர்களை ஒரே பயன்பாட்டில் வேலை செய்ய அனுமதிக்கிறது.
- பல கோப்புகளில் இந்த குறிமுறை சேமிக்கப்படுகிறது.
- குறிமுறை சிறியதாக, எளியதாக, புரிந்து கொள்ளும் வகையில் உள்ளது.
- துணை நிரல்களாக அல்லது செயல்கூறுகளாக இருப்பதால் பிழைகளை எளிதாக கண்டு பிடிக்க இயலும்.
- ஒரே குறிமுறை பல பயன்பாடுகளில் பயன்படுத்தப்படலாம்.
- மாறிகளின் வரையெல்லையை எளிதில் கட்டுப்படுத்த முடியும்.

### 3.5.3 அணுகல் கட்டுப்பாடு

அணுகல் கட்டுப்பாடு என்பது கணினி சூழலில் உள்ள வளங்களை யாரெல்லாம் பார்வையிட மற்றும் பயன்படுத்த முடியும் என்பதை வரைமுறைப்படுத்தும் ஒரு பாதுகாப்பு தொழில்நுட்பமாகும். இது பாதுகாப்பின் ஒரு அடிப்படைக் கருதாகும். பொருளுக்கான ஆபத்தைக் குறைக்கிறது. அதாவது அணுகல் கட்டுப்பாடு என்பது தரவை அணுகுவதற்கான குறிப்பிடப்பட்ட கட்டுப்பாடாகும். பொருள் நோக்கு நிரலாக்க மொழியில் இது அணுகியல்பு வரையறுப்புகள் மூலம் செயல்படுத்தப்படுகிறது. C++ மற்றும் Java போன்ற பொருள் நோக்கு மொழிகள் private, protected, public என்னும் சிறப்புச் சொற்களைப் பயன்படுத்தி இனக்குழு உறுப்புகளின் அணுகலைக் கட்டுப்படுத்துகிறது. private உறுப்புகளை இனக்குழுவிற்கு வெளியே இருந்து அணுக முடியாது. இனக்குழுவிற்கு உள்ளே மட்டும்தான் கையாள முடியும்.

public உறுப்புகளை இனக்குழுவிற்கு வெளியே இருந்தும் அணுக முடியும். public வழிமுறையை செயலாக்க அந்த இனக்குழுவின் பொருள் தேவைப்படுகிறது. private தரவுகள் மற்றும் public வழி முறைகளுக்கான இந்த ஏற்பாடு உறை பொதியாக்க கொள்கையை உறுதிப்படுத்துகிறது.

protected உறுப்புகள் அந்த இனக்குழு மற்றும் அதன் துணை இனக்குழுக்களால் அணுகப்படலாம். இதை அணுகுவதற்கு வேறு எந்த செயல்முறையும் அனுமதிக்கப்படாது. இது (Child) இனக்குழுவின் மரபுரிமை பெற்ற Parent இனக்குழுவின் குறிப்பிட்ட வளங்களை செயல்படுத்துகிறது. மாறி அல்லது வழிமுறையின் அணுகலை திறம்பட கட்டுப்படுத்துவதற்கான எந்த விதமான வழிமுறையையும் பைத்தான் கொண்டிருக்கவில்லை.

பைத்தான் ஒரு மாறி அல்லது வழிமுறையின் பெயருக்கு முன்னே ஒற்றை மற்றும் இரட்டை அடிக்கோடிடும் வழக்கத்தைப் பரிந்துரைக்கிறது. இதனால் Private மற்றும் Protected அணுகியல்பு வரையறுப்பிகள் சில பண்புகளைப் பின்பற்றுகின்றன.

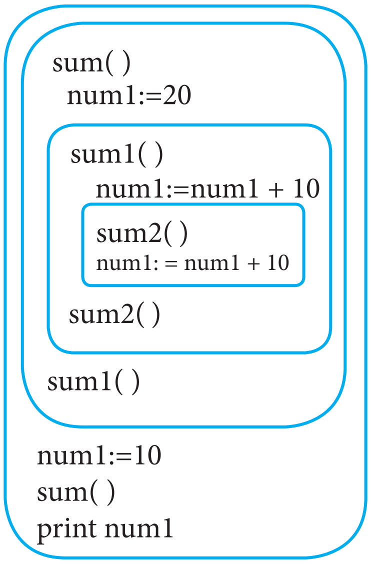
பைத்தானில் தானமைவாக இனக்குழுவின் அனைத்து உறுப்புகளும் public உறுப்புகளாகவும், C++ மற்றும் Java-வில் தானமைவாக private உறுப்புகளாகவும் உள்ளன. பைத்தானில் அனைத்து உறுப்புகளையும் இனக்குழுவிற்கு வெளியில் இருந்து அணுகமுடியும். ஆனால், C++ மற்றும் Java-வில் இவ்வாறு அணுக முடியாது.

## நினைவில் கொள்க

- வரையெல்லை என்பது மாறிகள், அளபுருக்கள் மற்றும் செயற்கூறுகளின் அணுகியல்பை நிரலின் ஒரு பகுதியில் இருந்து மற்றொரு பகுதிக்கு குறிப்பதாகும்.
- மாறியின் பெயரை ஒரு பொருளுடன் பிணைக்கும் செயல்முறையே மேப்பிங் (Mapping) எனப்படும். = செயல்குறி நிரலாக்க மொழியில் மாறி மற்றும் பொருளை மேப் செய்கிறது.
- namespaces என்பது மாறியின் பெயரை பொருளுடன் மேப்பிங் செய்வதற்கான இடம்.
- மாறியின் வரையெல்லை என்பது அது குறிமுறையில் எங்கு புலப்படுகிறதோ அல்லது காணப்படுகிறதோ அந்தப் பகுதியாகும்.
- LEGB விதி வரையெல்லை தேடப்பட வேண்டிய வரிசையை தீர்மானிக்கப் பயன்படுகிறது.
- உள்ளமை வரையெல்லை நடப்பு செயற்கூறில் வரையறுக்கப்பட்ட மாறிகளைக் குறிக்கும்.
- நிரலின் அனைத்து செயற்கூறுகளுக்கும் வெளியே அறிவிக்கப்பட்ட மாறிகள் முழுதளாவிய மாறிகள் எனப்படும்.
- ஒரு செயற்கூறின் உள்ளே மற்றொரு செயற்கூறு அடைக்கப்பட்டிருந்தால் அது பின்னலான செயற்கூறு எனப்படும்.
- மற்றொரு செயற்கூறு வரையறையை, தன்னுள் கொண்ட ஒரு வெளி செயற்கூறினுள் ஒரு மாறி அறிவிக்கப்பட்டால் உள்செயற்கூறானது, வெளி செயற்கூறினுள் உள்ள மாறிகளை அணுக முடியும். இதுவே, இணைக்கப்பட்ட வரையெல்லை (Enclosed Scope) எனப்படும்.
- நிரல் பெயர்ப்பி அல்லது தொகுப்பானை தொடங்கம் பொழுது உள்ளிணைந்த வரையெல்லையானது நிரல் வரையெல்லையில் ஏற்கனவே கொடுக்கப்பட்ட அனைத்து பெயர்களையும் கொண்டிருக்கும்.
- நிரலின் ஒரு பகுதியே தொகுதியாகும். நிரல்கள் ஒன்று அல்லது அதற்கு மேற்பட்ட தனித்து உருவாக்கப்பட்ட தொகுதிகளால் அமைக்கப்படுகிறது.
- ஒரு கணிப்பொறி நிரலை பல துணை நிரல்களாக பிரிக்கும் செயல்முறையே தொகுதி நிரலாக்கம் எனப்படும்.
- அணுகல் கட்டுப்பாடு என்பது கணினி சூழலில் உள்ள வளங்களை யாரெல்லாம் பார்வையிட மற்றும் பயன்படுத்த முடியும் என்பதை வரைமுறைப்படுத்தும் ஒரு பாதுகாப்பு தொழில் நுட்பமாகும். இது பாதுகாப்பின் ஒரு அடிப்படைக் கருத்தாகும். பொருளுக்கான ஆபத்தை குறைக்கிறது.
- public உறுப்புகளை இனக்குழுவிற்கு வெளியே இருந்தும் அணுக முடியும்.
- protected உறுப்புகள் அந்த இனக்குழு மற்றும் அதன் துணை இனக்குழுக்களால் அணுகப்படலாம்.
- private உறுப்புகளை இனக்குழுவிற்கு வெளியே இருந்து அணுக முடியாது. இனக்குழுவிற்கு உள்ளே மட்டும்தான் கையாள முடியும்.
- பைத்தான் ஒரு மாறி அல்லது வழிமுறையின் பெயருக்கு முன்னே ஒற்றை மற்றும் இரட்டை அடிக்கோடிடும் வழக்கத்தைப் பரிந்துரைக்கிறது. இதனால் private மற்றும் protected அணுகியல்பு வரையறுப்பிகள் சில பண்புகளைப் பின்பற்றுகின்றன.
- C++ மற்றும் Java போன்ற பொருள் நோக்கு மொழிகள் private, public, protected என்னும் சிறப்புச் சொற்களைப் பயன்படுத்தி இனக்குழு உறுப்புகளின் அணுகலைக் கட்டுப்படுத்துகிறது.
- பைத்தானில் தானமைவாக இனக்குழுவின் அனைத்து உறுப்புகளும் public உறுப்புகளாகவும், C++ மற்றும் Javaவில் தானமைவாக private உறுப்புகளாகவும் உள்ளன.

## செய்முறைப் பயிற்சி

1. கீழே கொடுக்கப்பட்டுள்ள படத்திற்கு ஏற்ற போலி குறிமுறையை எழுதுக.



```
sum( )
  num1:=20
  sum1( )
    num1:=num1 + 10
    sum2( )
      num1: = num1 + 10
    sum2( )
  sum1( )
num1:=10
sum( )
print num1
```

KRVCK1

## மதிப்பீடு

### பகுதி–அ

**சரியான விடையை தேர்ந்தெடுத்து எழுதுக** **(1 மதிப்பெண்)**

1. பின்வருவனவற்றுள் எது நிரலின் ஒரு பகுதியின் அணுகியல்பை மற்றொரு பகுதிக்கு குறிப்பதாகும்?

    (அ) வரையெல்லை (ஆ) நினைவகம் (இ) முகவரி (ஈ) அணுகுமுறை

2. மாறியின் பெயரை ஒரு பொருளுடன் பிணைக்கும் செயல்முறையை என்னவென்று அழைக்கப்படும்?

    (அ) வரையெல்லை (ஆ) மேப்பிங் (இ) பின் பிணைத்தல் (ஈ) முன் பிணைத்தல்

3. பின்வருவனவற்றுள் எது நிரலாக்க மொழியில் மாறியையும் பொருளையும் மேப் செய்யப் பயன்படுகிறது?

    (அ) :: (ஆ) := (இ) = (ஈ) ==

4. எது மாறியின் பெயரை பொருளுடன் மேப்பிங் செய்வதற்கான இடம் ஆகும்.

    (அ) வரையெல்லை (ஆ) மேப்பிங் (இ) பிணைத்தல் (ஈ) Namespaces

5. எந்த வரையெல்லை நடப்பு செயற்கூறில் வரையறுக்கப்படும் மாறிகளைக் குறிக்கும்?

    (அ) உள்ளமை வரையெல்லை (ஆ) முழுதளாவிய வரையெல்லை

    (இ) தொகுதி வரையெல்லை (ஈ) செயற்கூறு வரையெல்லை

6. ஒரு கணிப்பொறி நிரலை பல துணை நிரல்களாக பிரிக்கும் செயல்முறையே என்னவென்று அழைக்கப்படும்.

(அ) செயல்முறை நிரலாக்கம் (ஆ) தொகுதி நிரலாக்கம்

(இ) நிகழ்வு இயக்க நிரலாக்கம் (ஈ) பொருள் நோக்கு நிரலாக்கம்

7. எது கணினி சூழலில் உள்ள வளங்களை யார் பார்வையிட மற்றும் பயன்படுத்த முடியும் என்பதை வரைமுறைப்படுத்தும் ஒரு பாதுகாப்பு தொழில்நுட்பமாகும்.

(அ) கடவுச் சொல் (ஆ) அங்கீகாரம்

(இ) அணுகல் கட்டுப்பாடு (ஈ) சான்றிதழ்

8. எந்த இனக்குழுவின் உறுப்புகளை இனக்குழுவின் உள்ளே மட்டும்தான் கையாள முடியும்.

(அ) public உறுப்புகள் (ஆ)protected உறுப்புகள்

(இ) pecured உறுப்புகள் (ஈ) private உறுப்புகள்

9. எந்த உறுப்புகளை இனக்குழுவிற்கு வெளியே இருந்தும் அணுக முடியும்?

(அ) public உறுப்புகள் (ஆ) protected உறுப்புகள்

(இ) pecured உறுப்புகள் (ஈ) private உறுப்புகள்

10. எது வரையறுக்கப்பட்ட இனக்குழு மற்றும் அதன் துணை இனக்குழுக்களால் அணுகப்படும் உறுப்புகள் ஆகும்.

(அ) public உறுப்புகள் (ஆ) protected உறுப்புகள்

(இ) pecured உறுப்புகள் (ஈ) private உறுப்புகள்

## பகுதி–ஆ

**அனைத்து வினாக்களுக்கும் விடையளி** **(2 மதிப்பெண்)**

1. வரையெல்லை என்றால் என்ன?
2. மாறிகளுக்கு எதற்காக வரையெல்லை பயன்படுத்தப்பட வேண்டும்? காரணம் கூறுக?
3. மேப்பிங் என்றால் என்ன?
4. Namespaces சிறுகுறிப்பு வரைக?
5. private மற்றும் protected அணுகியல்புகளை பைத்தான் எவ்வாறு குறிப்பிடுகிறது.

## பகுதி–இ

**அனைத்து வினாக்களுக்கும் விடையளி** **(3 மதிப்பெண்)**

1. உள்ளமை வரையெல்லையை எடுத்துக்காட்டுடன் விவரி?
2. முழுதளாவிய வரையெல்லையை எடுத்துக்காட்டுடன் விவரி?
3. அடைக்கப்பட்ட வரையெல்லையை எடுத்துக்காட்டுடன் விவரி?
4. அணுகல் கட்டுப்பாடு எதற்குத் தேவைப்படுகிறது?

5. பின்வரும் போலிக் (Pseudo) குறிமுறையில் மாறிகளின் வரையெல்லையைக் கண்டறிந்து வெளியீட்டை எழுதுக.

```
output
color:= Red
mycolor():
  b:=Blue
  myfavcolor():
      g:=Green
      printcolor, b, g
  myfavcolor()
  printcolor, b
mycolor()
print color
```

## பகுதி ஈ

**அனைத்து வினாக்களுக்கும் விடையளி** **(5 மதிப்பெண்)**

1 மாறியின் வரையெல்லைகளின் வகைகளை விளக்குக (அல்லது) LEGB விதியை எடுத்துக்காட்டுடன் விளக்குக?

2. தொகுதிகளின் ஐந்து பண்பியல்புகளை எழுதுக?

3. தொகுதி நிரலாக்கத்தின் பயன்களை எழுதுக?

## REFERENCES

1. *Data Structures and Algorithms in Python By Michael T.Goodrich, RobertoTamassia and Michael H. Goldwasser.*
2. *Data Structure and Algorithmic Thinking in Python By Narasimha Karumanchi*
3. *https://www.python.org*