அலகு V

பாடம் 14

பைத்தானில் C++ நிரல்களை இறக்கம் செய்தல்

XNH79W

கற்றலின் நோக்கங்கள்

இந்த பாடப்பகுதியை கற்றபின் மாணவர்கள பின்வருவனவற்றை அறிந்து கொள்வார்கள்

- உறை இடுதல் (wrapping) என்றால் என்ன என்பதை புரிந்துக் கொள்ளுதல்.
- C++ செயற்கூறுகளையும் இனக்குழுக்களையும் பைத்தான் நிரல்களுக்குள் தருவித்து கொள்ளுதல்.
- இருநிரலாக்க மொழிகளிலும் செயலாக்கவல்ல சூழலை உருவாக்குதல்.
- பைத்தான் நிரல்களை இயக்குதல் மற்றும் பிழைத்திருத்தம் செய்தல்.

14.1 அறிமுகம்

பைத்தான் மற்றும் C++ ஒரு பொதுப்பயன் நிரலாக்க மொழியாகும். இருப்பினும் பைத்தான் C++ மொழியைக் காட்டிலும் முற்றிலும் மாறுபட்டது.

வ.எண்	பைத்தான்	C++
1	பொதுவாக, பைத்தான் ஒரு 'வரி மொழி மாற்றி' மொழியாகும்.	பொதுவாக C++ மொழி, ஒரு தொகுப்பு மொழியாகும்.
2	பைத்தான் மாறும் தன்மைக் கொண்ட மொழியாகும்.	C++ நிலையான தன்மைக் கொண்ட தொகுக்கப்பட்ட மொழி.
3	மாறியை அறிவிக்கும் போது, அது சார்ந்த தரவின வகையை குறிப்பிடத் தேவையில்லை	மாறியை அறிவிக்கும் போது, தரவினத்தை குறிப்பிட வேண்டும்.
4	இது Scripting மற்றும் பொதுப் பயன் மொழியென இருவகையிலும் செயல்பட முடியும்.	இது ஒரு பொதுப்பயன் நிரலாக்க மொழியாகும்.

ஆயினும், இவ்விரு மொழிகளும் ஒன்றுக்-கொன்று சரியான வகையில் பொருந்திக் கொள்ளும். பைத்தான் , பெரும்பாலும் scripting அல்லது "glue" மொழியாக பயன்படுத்தப்படுகிறது. அதாவது, மேல் நிலையிலுள்ள நிரல் பெரும்பாலும் C அல்லது C++ மொழிகளில் எழுதப்பட்டுள்ள செயல் முறைகளை அழைக்கும். இது தருக்க கருத்தை தற்போதைய குறிமுறையில் (எடுத்துக்காட்டாக C++ மொழியில் எழுதப்பட்டுள்ள நிரல்) எழுதும் போது உதவிகரமாக இருக்கும். ஆனால், பைத்தான் நிரலின் மூலம் அழைத்து, கையாளலாம்.

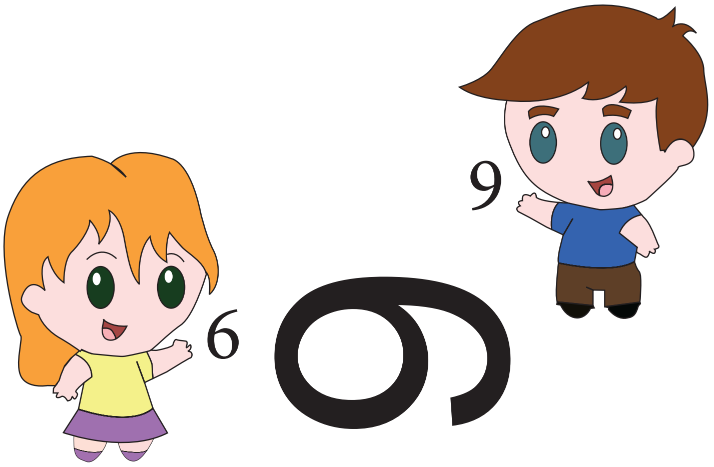
Chapter 14.indd 271 10-02-2020 20:04:00

14.2 Scripting மொழி

ஒரு scripting மொழி என்பது பிற நிரலாக்க மொழிகளுடன் ஒருங்கிணைப்பதற்கும், தொடர்பு கொள்வதற்கும் வடிவமைக்கப்பட்ட ஒரு நிரலாக்க மொழியாகும். ஜாவாஸ்கிரிப்ட், VBஸ்கிரிப்ட், PHP, பெர்ல், பைத்தான், ரூபி, ASP மற்றும் Tcl ஆகியவை மிகவும் பரவலாக பயன்படுத்தப்படும் Scripting மொழியாகும். பொதுவாக, ஒரு Scripting மொழி மற்ற நிரலாக்க மொழிகளுடன் இணைக்கப்பட்டு பயன்படுத்தப்படுவதால், பெரும்பாலும், அவை HTML, ஜாவா அல்லது C++ மொழிகளுடன் இணைந்து காணப்படுகின்றது.

14.2.1 ஸ்கிரிப்டிங் மற்றும் நிரலாக்க மொழிகளுக்கிடையே உள்ள வேறுபாடு

Scripting மொழி மற்றும் நிரலாக்க மொழி கீழ்க்காணும் வரைப்படத்தைப் போன்று காணப்படும்.



அடிப்படையில், அனைத்து Scripting மொழிகளும் நிரலாக்க மொழிகளாகும். இரண்டுக்குமிடையே தத்துவார்த்தமான வேறுபாடு என்னவெனில், Scripting மொழிக்கு தொகுத்தல் படிநிலைத் தேவைப்படாது, மாறாக விளக்கம் தேவைப்படும்.. உதாரணத்திற்கு, ஒரு C++ நிரல் இயக்குவதற்கும் முன்பாக, தொகுக்கப்பட வேண்டும். அதே சமயம் ஜாவாஸ்கிரிப்ட் அல்லது பைத்தான் போன்ற Scripting மொழிகள் தொகுக்கப்பட்ட தேவையில்லை. நிரலாக்க மொழிக்கு தொகுப்பான் / நிரல்பெயர்ப்பி தேவைப்படுவது போல Scripting மொழிக்கு வரி மொழி மாற்றித் தேவைப்படுகிறது. கொடுக்கப்பட்ட மொழி நிரலாக்க மொழியா அல்லது Scripting மொழியா என்பது அவை பயன்படுத்தப்படும் சூழலைப் பொறுத்தது.

14.3 Scripting மொழிகளின் பயன்பாடுகள்

1. ஒரு நிரலில் சில செயல்பாடுகளை தானியங்குப்படுத்துதல்.
2. தரவு தொகுப்பிலிருந்து தகவலைப் பிரித்தெடுத்தல்.
3. பழமையான நிரலாக்க மொழிகளுடன் ஒப்பிடும் போது, குறைந்த நிரல் குறிமுறையைக் கொண்டது.
4. பயன்பாடுகளுக்கு புதிய செயல்பாடுகளை கொண்டு வர முடியும். மேலும், சிக்கலான அமைப்புகளை ஒருங்கமைக்க முடியும்.

பைத்தான் என்பது உண்மையில், எந்த ஒரு தற்போதைய கணினி இயக்க அமைப்பிலும் பயன்படுத்தக்கூடிய ஒரு வரிமொழி மாற்று, உயர்நிலை, பொது பயன்பாட்டு, நிரலாக்க மொழி ஆகும். இது உரை, எண்கள், படங்கள் மற்றும் அறிவியல் சார்ந்த தரவுகளைச் செயலாக்கப்பயன்படுகிறது. தற்போது, பெரும்பான்மையான, பெரிய பயன்பாடுகள் பிரத்யேகமாக பைத்தான் மொழியில் எழுதப்படுகின்றன.

Chapter 14.indd 272 10-02-2020 20:04:01

14.4 C++மீது பைத்தான் பண்புக்கூறுகள்

- பைத்தான் தேவையற்ற (garbage) மதிப்புகளைச் சேகரிக்கும் தானியங்கியைப் பயன்படுத்துகிறது.. இந்த பண்புக்கூறு C++ ல் கிடையாது.
- C++ நிலையான வகையைச் சார்ந்த மொழி, ஆனால் பைத்தான் ஒரு மாறக்கூடிய வகையைச் சார்ந்த மொழியாகும்.
- பைத்தான் வரி மொழி மாற்றி மூலம் இயங்குகிறது. ஆனால் C++ மொழி முன் தொகுக்கப்பட்டது.
- C++ நிரல் குறிமுறையைக் காட்டிலும் பைத்தான் குறிமுறை 5லிருந்து 10 தடவைகள் (மடங்குகள்) குறைவானது.
- பைத்தானில், வெளிப்படையாக தரவினங்களை அறிவிக்க தேவையில்லை. ஆனால் C++-ல் அவை அறிவிக்கப்பட வேண்டும்.
- பைத்தானில், ஒரு செயற்கூறு எந்த வகை செயலுருபையும் ஏற்கும். மேலும், முன்னதாக எந்த ஒரு அறிவிப்பும் இல்லாமல் பல மதிப்புகளை திருப்பியனுப்பும். ஆனால் C++ return கூற்று ஒரே ஒரு மதிப்பை மட்டுமே திருப்பியனுப்பும்.

குறிப்பு

பைத்தான், நினைவக இடத்தை விடுவிக்க, தேவையற்ற பொருட்களை (உள்ளமைக்கப்பட்ட வகைகள் அல்லது இனக்குழு சான்றுருக்கள்) தானமைவாக நீக்குகிறது.அவ்வப்போது, பைத்தான், நினைவகப் பகுதிகளை விடுவித்து, பயன்பாட்டில் இல்லாதவைகளை மீட்டெடுக்கும் செயல்முறை தேவையற்ற மதிப்புகளை சேகரித்தல் (Garbage collection) என்றழைக்கப்படுகிறது.

14.5 பைத்தானில் C++ கோப்புகளைத் தருவித்துக் கொள்ளுதல்

பைத்தான் நிரலில் C++ நிரலை தருவித்துக் கொள்ளுதலைப் பைத்தானில் C++ -யை உறைஇடுதல் என்கிறோம். C++ நிரல்களுக்கான பைத்தான் இடைமுகங்களை அல்லது உறை இடுதலை பல வழிகளில் உருவாக்கலாம். பொதுவாக பயன்படுத்தப்படும் இடைமுகங்களாவன

- Python-C-API (API-Application Programming Interface C நிரல்களுடன் தொடர்பு கொள்ள)
- Ctypes (c நிரல்களுடன் தொடர்பு கொள்ள)
- SWIG (Simplified Wrapper Interface Generator- C மற்றும் C++ இரண்டு மொழிகளுக்கும்)
- Cython (C-நீட்டிப்புக்களை எழுதுவதற்கான ஒரு பைத்தான் போன்ற மொழியாகும்.)
- Boost. Python (Python மற்றும் C++ தொடர்பு கொள்வதற்கான கட்டமைப்பு)
- MinGW(விண்டோஸ்-க்கான குறைந்த பட்ச GNU)

14.5.1 MinGW இடைமுகம்

MinGW இயக்க நேர தலைப்புக் கோப்புகளின் தொகுப்பைக் குறிக்கிறது. விண்டோஸ் இயக்க முறைமையில் இயங்கும் வகையில், C , C++ மற்றும் FORTRAN நிரல் குறிமுறைகளைத் தொகுக்கவும், இணைக்கவும் இது பயன்படுகிறது.

MGPVNN

MinGw-W64 (MinGW இன் பதிப்பு) windows இன் C++ க்கு சிறந்த தொகுப்பான்/ நிரல் பெயர்ப்பி ஆகும். C++ நிரல்களை தொகுத்து, இயக்க, விண்டோஸ் இயக்க முறைமைக்கு 'g++' தொகுப்பான் தேவை MinGW g++ஐ பயன்படுத்தி பைத்தான் நிரல் மூலம் C++ நிரல்களை தொகுத்து, இயக்க அனுமதிக்கிறது.

C++ நிரல் குறிமுறையைக் கொண்ட பைத்தான் நிரல்களை minGW-w64 project திட்டப்பணி (run முனையம்) மூலம் மட்டுமே இயக்க முடியும். பைத்தான் நிரல்கள் இயக்கப்பட வேண்டிய கட்டளை வரி சாளரத்தை **run முனையம்** திறந்து வைக்கும்.

> g++ என்பது GCC- யை(GNU C தொகுப்பான்) அழைக்கும் நிரல்.இது தானாகவே தேவையான C++ நூலக கோப்புகளை இலக்கு நிரலுடன் இணைக்கிறது.

Refer installation of MinGW in Annexure -2

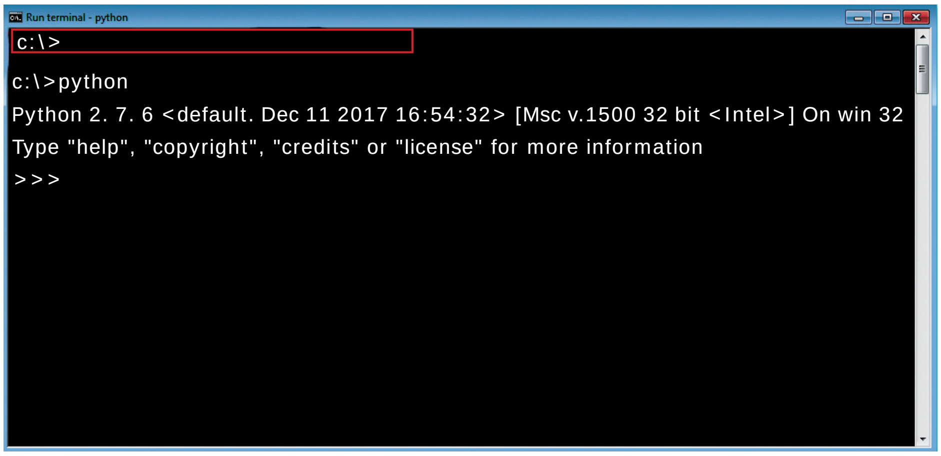
14.5.2 C++ நிரலை பைத்தான் மூலம் இயக்குதல்

1. MinGW run-ன் முனையத்தை இரட்டைக் கிளிக் செய்யவும்
2. பைத்தான் மென்பொருள் அமைந்திருக்கும் (python.exe) கோப்புறைக்கு செல்லவும். இந்த உதாரணத்தில், பைத்தான் கோப்புறை C:\Program Files\OpenOffiice4\Python என்ற அமைவிடத்தில் காணலாம்.

   c:\> கோப்புறையிலிருந்து பைத்தான் அமைந்திருக்கும் கோப்புறைக்கு மாறுவதற்கான தொடரியல்

   cd <absolute path>

இதில் "cd" கட்டளை change directory என்பதையும் absolute path என்பது பைத்தான் நிறுவப்பட்டிருக்கும் முழுமையான பாதையையும் குறிக்கும்.



படம் *14.5.2.1 -11*

இந்த எடுத்துக்காட்டில், பைத்தான் கோப்புறை அமைந்துள்ள இடத்திற்கு சென்று, பின்வரும் கட்டளையைத் தட்டச்சு செய்ய வேண்டும். "cd C:\>". மேற்கண்ட சாளரத்தில் உயர்த்திக்காட்டிருப்பதைக் காண்க.

உதாரணத்திற்கு, Pali.cpp என்ற C++ நிரலை படிக்கும் pycpp.py என்ற பைத்தான் நிரலை எடுத்துக் கொள்ளுங்கள். Pali.cpp என்ற நிரல் ஓர் எண்ணைப் பெற்றுக் கொண்டு, அது பாலிண்ட்ரோமா இல்லையா என்பதை வெளியிடும். உதாரணத்திற்கு, 232 என்ற எண்ணை உள்ளீடாக கொடுத்தால், palindrome என்ற வெளியீடு கிடைக்கும். C++ நிரல் pali நோட்பேடில் தட்டச்சு செய்யப்பட்டு, pali.cpp என்ற பெயரில்

Chapter 14.indd 274
10-02-2020 20:04:01

சேமிக்கப்படும். அதைப் போன்றே, pycpp.py என்று பைத்தான் நிரல் குறிமுறையும் நோட்பேடில் தட்டச்சு செய்யப்பட்டு, pycpp.py என்று சேமிக்கப்படுகிறது.

3. நம் நிரலை இயக்க, run முனையத்தை இரட்டை கிளிக் செய்யவும். பாதையை பைத்தான் கோப்புறை அமைவிடத்திற்கும் மாற்றவும். பைத்தானை இயக்குவதற்கான தொடரியல்,

Python <filename.py> -i <C++ filename without cpp extension>

இதில்,

Python	கட்டளை வரியிலிருந்து பைத்தான் நிரலை செயல்படுத்துவதற்கான சிறப்புச் சொல்
filename.py	செயல்படுத்த வேண்டிய பைத்தான் நிரலின் பெயர்
- i	உள்ளீட்டு முறைமை
C++ filename without cpp extension	தொகுக்கப்பட்டு செயல்படுத்தப்பட வேண்டிய C++ நிரலின் பெயர்.

உதாரணத்திற்கு, கட்டளை தூண்டுக்குறியில், "Python pycpp.py –i pali" என்று உள்ளிட்டு enter விசையை அழுத்தவும். தொகுத்தல் வெற்றிகரமாக முடிந்தால் நீங்கள் எதிர்பார்த்த வெளியீடு கிடைக்கும் இல்லையேல் பிழைச் செய்தி வெளியிடப்படும்.

குறிப்பு

இயங்கு (execution) கட்டளையில், உள்ளீட்டு கோப்பிற்கு நீட்டிப்பு தேவைப்படாது. உதாரணத்திற்கு, "pali.cpp" என்பதற்கு பதிலாக "pali" என்ற பெயரைக் கொடுத்தால் போதுமானது.

நாம் pycpp.py மற்றும் pali.cpp என்ற எடுத்துக்காட்டுகள் மூலம் நிரலை இயக்குதல் பற்றி பார்ப்போம். இந்த இரு நிரல்களும் c:\pyprg என்ற கோப்புறையில் இருத்தி வைக்கப்பட்டுள்ளன. இந்த நிரல்கள் பைத்தானில் "exe file" அமைந்திருக்கும் அதே கோப்புறையில் இல்லாவிடில், இயக்க நேரத்தின் போது கோப்புகளின் முழுமையான பாதையைக் குறிப்பிடுதல் அவசியம். வெளியீடு கீழ்க்கண்டவாறு இருக்கும்.

```
Run terminal
C:\Program Files\migw-w64\i686-8.1.0-posix-dwarf-rt_v6-rev0>echo off
Microsoft windows [Version 6.1.7601]
copy right <c> 2009 Microsoft Corporation. All rights reserved.

C:\>cd program files\openoffice 4 program

C:\Program Files\openoffice 4\program> Python c:\pyprg\pycpp.py-i c:\pyprg\pali
compailing c:\pyprg\pali.cpp
Running c:\pyprg\pali.exe
----------------------
Enter apositive number:232
  The reverse of the number is:232
  The number is a palindrome
C:\Program jFiles\Openoffice 4\program>Python c:\pyprg\pycpp.py -i c:\pyprg\pali
Compiling c:\pyprg\pali.cpp
Running c:\pyprg\pali.exe
----------------------
Enter a positive number:234
  The reverse of the number is:432
  The number is not a palindrome
C:\Program Files\Openoffice 4\program>_
```

படம்14.2

Chapter 14.indd 275 10-02-2020 20:04:01

குறிப்பு

கட்டளை சாளரத்தில் உள்ள திரையை அழிக்க (clear) செய்ய cls கட்டளையைப் பயன்படுத்தவும்.

நாம் இப்பொழுது C++ நிரல் குறிமுறையைத் தொகுப்பதற்கான பைத்தான் நிரலை எவ்வாறு எழுதுவது என்பதைப் பார்ப்போம்.

14.6 C++ நிரலைத் தருவித்துக் கொள்வதற்கான பைத்தான் நிரல்

பைத்தான் பல கூறுநிலைகளைக் கொண்டுள்ளது. ஒரு சிக்கலுக்கு நிரலர்கள் பல தரப்பட்ட கூறுநிலைகளைத் தங்கள் வசதிக்கேற்ப பயன்படுத்திக் கொள்ள பைத்தான் அனுமதிக்கிறது. இப்பொழுது நாம் கற்றுக் கொள்ளவிருக்கும் பைத்தான் நிரல், அடிப்படை பைத்தான் நிரலில் நீங்கள் அறிந்திராத சில புதிய கட்டளைகளைக் கொண்டுள்ளது. இந்த அதிகாரத்தில், நிரல்கள் இரண்டு வெவ்வேறு மொழிகளின் ஒருங்கிணைப்பைப் பெற்றுள்ளதால், os,sys மற்றும் getopt போன்ற கூறுநிலைகளைத் தருவித்துக் கொள்ள வேண்டும்.

14.6.1 கூறுநிலை

கூறுநிலை நிரலாக்கம் என்பது உங்கள் குறிமுறையை சிறுசிறு பகுதிகளாக பிரிப்பதற்கான மென்பொருள் வடிவமைப்பு நுட்பமாகும். இந்த பகுதிகள் கூறுநிலைகள் என்றழைக்கப்படுகின்றன. இந்த பிரித்தலின் போது கூறுநிலைகள் என்றழைக்கப்படுகிறது. இந்த பிரித்தலின் போது கூறுநிலைகள் மற்ற பிற கூறுநிலைகளின் மீது குறைந்த அளவிலோ அல்லது இல்லாமாலோ சார்ந்திருப்பதை கவனத்தில் கொள்ள வேண்டும். வேறுவகையில், கூறவேண்டுமெனில் சார்பு நிலையை குறைத்தலே நோக்கமாகும்.

ஆனால், நாம் பைத்தானில் கூறுநிலைகளை எவ்வாறு உருவாக்குவோம்? பைத்தானில் கூற்றுகள் மற்றும் வரையறைகளைக் கொண்டுள்ள கோப்பு கூறுநிலை எனப்படுகிறது. பைத்தான் குறிமுறையை கொண்டிருக்கும் கோப்பு எடுத்துக்காட்டாக, factorial.py என்பதனை கூறுநிலை என்றும், factorial என்பது கூறுநிலையின் பெயரைக் குறிக்கும். நாம் கூறுநிலைகளைக் கையாளக்கூடிய, ஒருங்கமைக்கப்பட்ட சிறிய கோப்புகளாக பிரித்து பயன்படுத்துகிறோம். மேலும், கூறுநிலை, குறிமுறையின் மறுபயனபாக்கத்தை வழங்குகிறது. வெவ்வேறு நிரல்களில், நாம் மிகுதியாக பயன்படுத்தும் செயற்கூறு வரையறைகளை நகலெடுப்பதற்கு பதிலாக, அவற்றை ஒரு கூறுநிலையில் வரையறுத்து, தருவித்துக் கொள்ளலாம்.

எடுத்துக்காட்டு

```
def fact(n):
        f=1
        if n == 0:
        return 0
elif n == 1:
        return 1
        else:
                for i in range(1, n+1):
                        f= f*i
                print (f)
```

வெளியீடு

```
        >>>fact (5)
        120
```

மேற்கண்ட எடுத்துக்காட்டு factorial.py என பெயரிடப்பட்டுள்ளது.

Chapter 14.indd 276 10-02-2020 20:04:01

14.6.2 பைத்தானில் கூறுநிலைகளை எவ்வாறு தருவித்துக் கொள்வது?

ஒரு கூறுநிலைக்குள் மற்றொரு கூறுநிலையின் வரையறைகளைத் தருவித்துக் கொள்ளலாம். அவ்வாறு செய்வதற்கு நாம் 'import' என்ற சிறப்பு சொல்லைப் பயன்படுத்த முடியும். நாம் முன்னதாக வரையறுத்த factorial என்ற கூறுநிலையைத் தருவித்துக் கொள்ள, பைத்தான் தூண்டுகுறியில் கீழ்க்கண்டவாறு தட்டச்சு செய்யவும்.

>>> import factorial

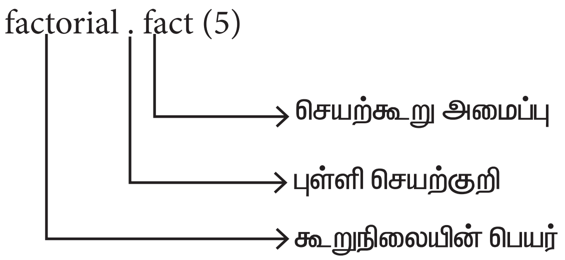
கூறுநிலைக்குள் வரையறுக்கப்பட்ட செயற்கூறுகளை, கூறுநிலையின் பெயரைப் பயன்படுத்தி அணுக முடியும். செயற்கூறுகளை அணுக புள்ளிச் செயற்குறி பயன்படுத்தப்படுகிறது கூறுநிலையிலிருந்து செயற்கூறுகளை அணுகுவதற்கான தொடரியல்

<module name> . <function name>

எடுத்துக்காட்டாக:

>>> factorial.fact(5)

120



பைத்தான் அடிப்படை (உள்ளிணைந்த) கூறுநிலைகள் பலவற்றைக் கொண்டுள்ளது. பயனர் வரையறுத்த கூறுநிலைகளைத் தருவிப்பது போன்றே அடிப்படை கூறுநிலைகளையும் தருவித்துக் கொள்ளலாம். நீங்கள் தற்போது C++ குறிமுறையை இயக்கும் பைத்தான் நிரலுக்கு தேவையான அடிப்படை கூறுநிலைகளைக் காணப் போகிறீர்கள்.

(i) பைத்தானின் sys கூறுநிலை

இந்த கூறுநிலை வரிமொழி மாற்றியால் பயன்படுத்தப்படுகிறது.

sys.argv

sys.argv என்பது பைத்தான் நிரலுக்கு அனுப்பி வைக்கப்படும் கட்டளை வரி செயலுருபுகளின் பட்டியலாகும். argv கட்டளை வரி உள்ளீட்டு வழியாக வரும் உருப்படிகள் அனைத்தையும் கொண்டிருக்கும். இது , அடிப்படையில், நிரலின் கட்டளை வரி செயலுருபுகளைக் கொண்ட ஓர் அணியாகும்.

sys.argv ஐ பயன்படுத்த, முதலில் நீங்கள் sys கூறுநிலையைத் தருவித்துக் கொள்ள வேண்டும். முதல் செயலுருபு sys.argv[0] எப்பொழுதும் செயல்படுத்த வேண்டிய நிரலின் பெயராக இருக்கும். மேலும், sys.argv[1] என்பது நிரலுக்கு (இங்கு இது C++கோப்பு) அனுப்பப்படும் முதல் செயலுருபு ஆகும்.

main(sys.argv[1])	நிரல் கோப்பையும் (பைத்தான் நிரல்), உள்ளீட்டு கோப்பினை (C++ கோப்பு) பட்டியலாகவும் (அணி) பெற்றுக் கொள்கிறது. argv[0] அனுப்பப்பட தேவையில்லாத பைத்தான் நிரலைக் கொண்டிருக்கும். ஏனெனில் default-main மூலக் குறிமுறையைக் குறிக்கும். செயல்படுத்த வேண்டிய கோப்பின் பெயரைக் குறிக்கும்.

Chapter 14.indd 277 10-02-2020 20:04:01

(ii) பைத்தானில் OS கூறுநிலை

பைத்தானில் இருக்கும் OS கூறுநிலை இயக்க முறைமையை சார்பு செயல்பாட்டுடன் பயன்படுத்துவதற்கான ஒரு வழிமுறையை வழங்குகிறது.

பைத்தான் இயங்கிக் கொண்டிருக்கும்போது, விண்டோஸ் இயக்க முறைமையுடன் OS கூறுநிலை ஊடாட அனுமதிக்கும் செயற்கூறுகளாவன.

os.system() : செயல்தளத்தில்(Shell)C++ தொகுத்தலுக்கான கட்டளையை இயக்கும். (இங்கு இது ஒரு கட்டளை சாளரம்) உதாரணத்திற்கு, C++ நிரலைத் தொகுக்க g++ தொகுப்பி செயல்படுத்தப்பட வேண்டும்.

இதை செயல்படுத்த, பின்வரும் கட்டளைப் பயன்படுத்தப்படுகின்றது.

os.system ('g++' + <variable_name1> '-<mode>' + <variable_name2>

இங்கு,

os.system :-	OS கூறுநிலையில் வரையறுக்கப்பட்டுள்ள system() செயற்கூறு
g++ :-	Windows இயக்கமுறைமையில் C++ நிரலை தொகுப்பதற்கான பொதுவான தொகுப்பி
variable_name1:-	(.cpp) என்ற நீட்டிப்பு இல்லாமல் C++ கோப்பின் பெயர் சர வடிவமைப்பில் குறிப்பிடப்பட்டிருக்கும்.
mode :-	உள்ளீடு/வெளியீட்டு பாங்கினை குறிக்கிறது. இங்கு இது hypen முன்னொட்டுடன் 0 என்று உள்ளது.
variable_name2 :-	(.exe) என்ற நீட்டிப்பு இல்லாமல் இயக்குக் கோப்புப் பெயர் சர வடிவமைப்பில் குறிப்பிடப்பட்டிருக்கும்.

உதாரணமாக, C++ நிரலை தொகுத்து, இயக்குவதற்கான கட்டளை கீழே கொடுக்கப்பட்டுள்ளது.

os.system('g++ ' + cpp_file + ' -o ' + exe_file)	g++ தொகுப்பான் cpp_file என்ற கோப்பினைத் தொகுக்கும்.–o(output) send to exe_file

குறிப்பு

OS.system()ல் '+' என்பது அனைத்து சரங்களும் ஒரே சரமாக இணைக்கப்பட்டு பட்டியலாக அனுப்பப்படுகிறது என்பதைக் குறிக்கிறது.

(iii) பைத்தான் getopt கூறுநிலை

பைத்தானில் getopt கூறுநிலை கட்டளை வரி தேர்வுகளையும், செயலுருபுகளையும் பிரித்தெடுக்க உங்களுக்கு உதவும். இந்த கூறுநிலை கட்டளைவரி செயலுருபு பிரித்தெடுத்தலை செயல்படுத்த செயற்கூறுகளை வழங்குகிறது.

getopt.getopt வழிமுறை

இந்த வழிமுறை கட்டளை வரி தேர்வுகளையும், செயலுருபுகள் பட்டியலையும் பிரித்தெடுக்கும். இந்த வழிமுறைக்கான தொடரியல் பின்வருமாறு

Chapter 14.indd 278 10-02-2020 20:04:01

<opts>,<args>=getopt.getopt(argv, options, [long_options])

செயலுருபுகளின் விவரங்களை இங்கே காணலாம்.

argv – இது பிரிக்கப்பட வேண்டிய செயலுருபின் மதிப்புகளின் பட்டியலைக் குறிக்கும். நமது நிரலில், கட்டளை முழுமையும் பட்டியலாக அனுப்பப்படுகிறது.

options – இது பைத்தான் நிரல் உள்ளீடு அல்லது வெளியீட்டிற்கான தேர்வு எழுத்துக்களின் சரமாகும். இங்கு 'i' அல்லது 'o' போன்ற தேர்வுகளும், அதைத் தொடர்ந்து முக்காற்புள்ளியும் (:) அமைந்திருக்கும். இங்கு (:) பாங்கினை குறிப்பிடப் பயன்படுகிறது.

long_options – இந்த அளபுரு சரங்களின் பட்டியலுடன் செலுத்தப்படுகிறது. Long options-ன் செயலுருபைத் தொடர்ந்து ('=') என்ற சமக்குறி இடம்பெற வேண்டும். நம்முடைய நிரலில் C++ கோப்பின் பெயர் சரமாக செலுத்தப்படும். மேலும், அது ஒரு உள்ளீடு கோப்பு என்பதைக் குறிக்க, அதனுடன் 'i' என்ற தேர்வும் செலுத்தப்படும்.

getopt() method returns value consisting of two elements. வழிமுறை இரண்டு உறுப்புகளை கொண்டுள்ள மதிப்புகளை திருப்பியனுப்பும். இவை ஒவ்வொன்றும் தனித்தனியாக opts மற்றும் args என்ற இரண்டு வெவ்வேறு பட்டியலில் (அணிகள்) சேமிக்கப்படும். Opts பாங்கு,பாதைப் போன்ற பிரிக்கப்பட்ட சரங்களின் பட்டியலைக் கொண்டிருக்கும். Args, தவறான பாதை அல்லது பாங்கின் காரணமாக பிரிக்கப்பட முடியாத எந்தவொரு சரத்தின் பட்டியலைக் கொண்டிருக்கும். geopt() வழிமுறையின் மூலம் சரங்களை பிரித்தெடுக்கும் போது பிழையேதும் இல்லாவிட்டால் args வெற்று அணியாக அமையும்.

உதாரணத்திற்கு, p4 என்ற C++ கோப்பினை, கட்டளை வரியில் இயக்கப்போகும் பைத்தான் நிரல், பின்வரும் ஒன்றைப் போல getopt() வழிமுறையைக் கொண்டிருக்கும்.

opts, args = getopt.getopt (argv, "i:",['ifile='])

where **opts** contains	[('-i', 'c:\\pyprg\\p4')] என்பதைக் கொண்டிருக்கும்.
-i :-	தேர்வு பாங்கினைத் தொடர்ந்து இடம் பெற்றிருக்க வேண்டும்.
'c:\\pyprg\\p4'	மதிப்பு C++ கோப்பின் முழுமையான பாதை

நம்முடைய எடுத்துக்காட்டுகளில், கட்டளை வரி கட்டளைகளை அனைத்தும் பிரிக்கப்பட்டிருப்பதாலும், செயலுருபு மீதமில்லாமலிருப்பதாலும், இரண்டாவது செயலுருபு args வெறுமையாக இருக்கும் [ ]. Print() கட்டளையைப் பயன்படுத்தி args வெளிக்காட்டப்பட்ட செய்திருந்தால், வெளியீடு [ ] என இருக்கும்.

```
>>>print(args)
[]
```

குறிப்பு

பைத்தானில் அடிப்படை கூறுநிலைகளின் பட்டியலையும், அவை எதற்காக பயன்படுத்தப்படுகிறது என்பதையும் நீங்கள் சரிபார்த்துக் கொள்ள முடியும். இந்த கோப்புகள் பைத்தானை நீங்கள் நிறுவி இருக்கும் அமைவிடத்தில், Lib கோப்புறையில் இருக்கும் C++ நிரலை wrap செய்வதற்கான மேலும் சில கட்டளைகள்.

Chapter 14.indd 279 10-02-2020 20:04:01

Some more command for wrapping C++ code

```
if __name__=='__main__':
        main(sys.argv[1:])
```

__name__ (A Special variable) in Python

பைத்தானில் main() செயற்கூறு இல்லையென்பதால், வரிமொழி மாற்றிக்கு (interpreter) பைத்தான் நிரலை இயக்குவதற்கான கட்டளைக் கொடுக்கும்போது, 0 (பூஜ்யம்) நிலையில் 0 level உள்தள்ளப்பட்ட குறிமுறை இயக்கப்பட வேண்டும். இருப்பினும், வரிமொழி மாற்றி இதனை செய்வதற்கு முன், சில சிறப்பு மாறிகளை வரையறுத்துக்கொள்ளும். __name__ என்பது அத்தகைய சிறப்பு மாறி, தானமைவாக, கோப்பின் பெயரை இருத்திக் கொள்ளும். மூலக்கோப்பு முதன்மை நிரலாக இயக்கப்படுமேயானால், வரிமொழி மாற்றி, __name__ மாறிக்கு "__main__" என்ற மதிப்பினை அமைத்துக் கொடுக்கும்.

__name__ என்பது தற்போதைய கூறுநிலையை மதிப்பீடு செய்யும் ஒரு உள்ளிணைந்த மாறியாகும். இதனால் தற்போதைய script தானாகவே இயங்கி கொண்டிருக்கிறதா என்பதை சோதிக்கப்பயன்படுகிறது.

உதாரணத்திற்கு பின்வருவனவற்றை நோக்குங்கள்

```
if __name__ == '__main__':
        main (sys.argv[1:])
```

கட்டளை வரியில் பைத்தான் நிரல், தானே முதலில் செயல்பட போகிறதென்றால், __main__ அந்த பைத்தான் நிரலின் பெயரைக் கொண்டிருக்கும். மேலும், பைத்தானில் சிறப்பு மாறியான, __name__ என்பதும் பைத்தான் நிரலின் பெயரைக் கொண்டிருக்கும். மேலும், பைத்தானின் சிறப்பு மாறியான _name_ என்பதும் பைத்தான் நிரலின் பெயரைக் கொண்டிருக்கும். நிபந்தனை சரியெனில் அது main அழைக்கப்பட்டு, C++ கோப்பினை செயலுருபாக அனுப்பி வைக்கப்படும்.

குறிப்பு

sys.argv[1:] - பெருக்குப்பின் கோப்பு பெயர் அனைத்தையும் பெறவும்.
sys.argv[0:]-Script name (பைத்தான் நிரல்)
நீங்கள் பாடம் 8-ல் String slicing என்று படித்ததை நினைவில் கொள்ளுங்கள்

14.7 பாய்வு கட்டுப்பாட்டுகூற்றுகளை கொண்ட C++ நிரல்களை இயக்கும் பைத்தான் நிரல்

நாம் இப்பொழுது , ஒரு C++குறிமுறையை படித்து, இயக்கி, அதன் தீர்வைக் கண்டறிய ஒரு சோதனை நிரலை எழுதலாம். கொடுக்கப்பட்டுள்ள எண் பாலிண்ட்ரோமா இல்லையா என்பதைச் சோதிக்கும் C++ நிரலை இயக்குவதற்கான படிநிலைகள் கீழே தரப்பட்டுள்ளது.

Chapter 14.indd 280 10-02-2020 20:04:01

படிநிலை 1	உள்ளீடு செய்யப்பட்ட எண் பாலிண்ட்ரோமா இல்லையா என சோதிக்கும் C++ நிரலை நோடபேடில் தட்டச்சு செய்து **pali.cpp.cpp** என்று சேமிக்கவும்.
படிநிலை 2	பைத்தான் நிரலை தட்டச்சு செய்து pali.py என்று சேமிக்கவும்.
படிநிலை 3	Run முனையத்தை Run terminal கிளிக் செய்து கட்டளை சாளரத்தை திறக்கவும்.
படிநிலை 4	python.pali.py-ipali-cpp என்ற கட்டளையை தட்டச்சு செய்யவும்.

எடுத்துக்காட்டு:- 14.7.1 – ஏதேனும் ஓர் எண்ணை உள்ளீடு செய்து, while மடக்கினை பயன்படுத்தி அந்த எண் பாலிண்ட்ரோமா இல்லையா என்பதை கண்டறிய ஒரு C++ நிரலை எழுதுக.

```
/*. To check whether the number is palindrome or not using while loop.*/
//Now select File->New in Notepad and type the C++ program
#include <iostream>
using namespace std;
int main()
{
int n, num, digit, rev = 0;
cout<< "Enter a positive number: ";
cin>>num;
n = num;
    while(num)
    {   digit = num % 10;
        rev = (rev * 10) + digit;
        num = num / 10;    }
cout<< " The reverse of the number is: " << rev <<endl;
    if (n == rev)
        cout<< " The number is a palindrome";
    else
        cout<< " The number is not a palindrome";
    return 0;
}
// Save this file as pali_cpp.cpp
```

இப்பொழுது நோட்பேடில் *File->New* என்பதை தேர்ந்தெடுத்து நிரலை தட்டச்சு செய்யவும்.

கோப்பினை *pali.py* . என்று சேமிக்கவும். இது ஒரு *.cpp* கோப்பைத் தொகுத்து, இயக்குவதற்கான நிரல்

```
# Python c:\pyprg\pali.py -i c:\pyprg\pali_cpp
import sys, os, getopt
def main(argv):
    opts, args = getopt.getopt(argv, "i:")
    for o, a in opts:
        if o in "-i":
            run(a)
def run(a):
    inp_file=a+'.cpp'
    exe_file=a+'.exe'
    os.system('g++  ' + inp_file + '  -o  ' + exe_file)
    os.system(exe_file)
if __name__=='__main__':
        main(sys.argv[1:])
```

Output of the above program

Output 1

C:\Users\Dell>python c:\pyprg\pali.py -i c:\pyprg\pali_cpp

Enter a positive number: 56765

The reverse of the number is: 56765

The number is a palindrome

Output 2

C:\Users\Dell>python c:\pyprg\ch14\pali.py -i c:\pyprg\pali_cpp

Enter a positive number: 56756

The reverse of the number is: 65765

The number is not a palindrome

Chapter 14.indd 282 10-02-2020 20:04:01

பைத்தான் நிரல்	எப்படி வேலை செய்கிறது (அல்லது) வேலை செய்யும் விதம்
import sys, os, getopt	தேவையான செயற்கூறுகளை பயன்படுத்த sys, os மற்றும் getopt ஆகிய துணை நிரல்களை இணைத்துக்கொள்ள வேண்டும்.
def main(argv):	main( ) செயற்கூறு வரையறை. argv உள்ளீட்டு கோப்புநிலை மற்றும் C++ நிரல் கோப்பு ஆகியவற்றை ஒரு லிஸ்ட்-ஆக பெறும். எடுத்துக்காட்டு: ['i', 'C:\pyprg\pali_cpp']
opts, args = getopt. getopt(argv, "i:")	getopt( ), opts-ல் உள்ள கட்டளையை விருப்பத்தேர்வு மற்றும் செயலுருபு என இரண்டாக பிரிக்கின்றது. எடுத்துக்காட்டு: ['i', 'C:\pyprg\pali_cpp'] பிழை ஏதுமில்லை எனவே, 'args' [] என காட்டும்.
for o, a in opts:	'o' கோப்பு நிலையையும், 'a' என்பது C++ நிரலின் பாதையையும் கொண்டிருக்கும். அதாவது, print("o= ",o) என்பது o= -i எனவும், print("a= ", a) என்பது a= C:\pyprg\pali_cpp எனவும் காட்டும்.
if o in ("-i"):	o=='i' மெய்யா என சோதிக்கும்.
run(a)	C++ நிரலை அனுப்பி, run( ) அழைக்கப்படும்
def run():	run() செயற்கூறு வரையறை இங்கு தொடங்கிறது.
inp_file=a+'.cpp'	inp_file என்ற மாறி C++ நிரலை, .cpp என்ற நீட்சியுடன் இணைத்து பெற்றிருக்கும். எடுத்துக்காட்டு: print(inp_file) என்பது C:\pyprg\pali_cpp.cpp எனக்காட்டும்.
exe_file=a+'.exe'	exe_file என்ற மாறி, C++ நிரலை, .exe என்ற நீட்சியுடன் இணைத்து பெற்றிருக்கும். எடுத்துக்காட்டு: print(exe_file) என்பது C:\pyprg\pali_cpp.exe எனக்காட்டும்.
os.system('g++' + inp_file + '-o' + exe_file)	g++ நிரல் பெயர்ப்பி, inp_fileல் கொடுக்கப்பட்ட C++ நிரலை தொகுத்து, இயங்கு நிரலாக exe_fileல் சேமிக்கும்.
os.system(exe_file)	exe நிரல்கோப்பை இயக்கும்.
if __name__ == '__ main__':	__name__ பைத்தான் நிரலின் பெயரை சேமிக்கும். __main__ இந்த மாறியும், பைத்தான் நிரலின் பெயரை சேமிக்கும். இந்த இரண்டு மாறிகளின் மதிப்பும் நிகரானதாக இருப்பின்.
main(sys.argv[1:])	பைத்தான் நிரல் பெயரை தவிர்த்து, கட்டளை இடைமுகத்தின் வழியே செயலுருபை அனுப்பி, main( ) அழைக்கப்படும். argv[1:] என்பது -i C:\pyprg\pali_cpp எனப் பெற்றிருக்கும்.

பாலிண்ட்ரோம் நிரலில் பயன்படுத்தியதைப் போலவே இடமாற்று அணிக்கோவையை இயக்குவதற்கான பைத்தான் நிரல் இருப்பதை நீங்கள் கவனித்திருப்பீர்கள் இதிலிருந்து நீங்கள் புரிந்துக் கொண்டது என்ன?

இந்த பைத்தான் நிரல் முக்கியமாக 'i'/'o' போன்ற பாங்குகளுடன் கூடிய C++கோப்பை படிக்கப் பயன்படுத்தப்பட்டுள்ளது. இது கட்டளை வரியிலிருக்கும் ஒவ்வொரு மதிப்பையும் பிரித்தெடுத்து, "opts" எனப்படும் பட்டியலுக்கு அளபுருவாக அனுப்பி வைக்கும். "long option ifile" அதன் பாங்குடன், C++குறிமுறை முழுவதையும் படிப்பதற்கான குறிப்பாக அமைகிறது.

for மடக்கிலுள்ள பட்டியல் 'o' பாங்கு (i') ஐயும் 'a' எனும் மாறி, முழுமையான பாதையுடனான C++ நிரல் கோப்பையும் கொண்டிருக்கும். உதாரணத்திற்கு [<'c:\pyprg\cpp_file'>,' trans_cpp']) cpp/exe போன்ற நீட்டிப்டிப்புகள் பைத்தான் scriptனால் இணைக்கப்படுகின்றன.

__name__ variable பைத்தான் Script-ன் 'main' வரையறையிலிருந்து நிரலைத் தொடங்குமாறு அறிவுறுத்துகிறது. 'main' வரையறை நீட்டிப்புகளை பிரித்தெடுத்தல் மற்றும் இணைத்தல் செயல்பாடுகளை செய்கிறது. "run" வரையறை "g++" தொகுப்பானை செயல்படுத்தி, exe கோப்பை உருவாக்குறிது. "os" கூறுநிலையின் system() கட்டளை exe கோப்பை செயல்படுத்தி, விரும்பிய வெளியீட்டை பெற்றுத் தருகிறது. பைத்தான் Scriptனைக் கொண்டு கோப்பு நீட்புக்களை இணைத்துக் கொள்ள முடியுமாதலால், C நிரல்களையும் நாம் செயல்படுத்திக் கொள்ளலாம்.

14.8 பைத்தான் எவ்வாறு C++ நிரல்களின் பிழைகளைக் கையாள்கிறது

பைத்தான் வெற்றிகரமான C++ நிரலை இயக்குவது மட்டுமல்லாது, தொகுத்தலின் போது C++ கூற்றில் ஏதேனும் பிழைகள் இருப்பின் அவற்றை வெளியிடவும் உதவுகிறது. உதாரணத்திற்கு, கீழ்க்காணும் C++ நிரலின் ஒரு பிழை உள்ளது. பைத்தான் மூலம் தொகுக்கும் போது என்ன நேரும் என்பதைப் பார்க்கலாம்.

எடுத்துக்காட்டு 14.8.1

```
// C++ program to print the message Hello
//Now select File→New in Notepad and type the C++ program
#include<iostream>
using namespace std;
int main()
{
	std::cout<<"hello"
	return 0;
}
// Save this file as hello.cpp
```


Chapter 14.indd 284 10-02-2020 20:04:02

```
#Now select File→New in Notepad and type the Python program as main.py
# Program that compiles and executes a .cpp file
 # Python main.py -i  hello

import sys, os, getopt
def main(argv):
    opts, args = getopt.getopt(argv, "i:")
    for o, a in opts:
        if o in "-i":
            run(a)
def run(a):
    inp_file=a+'.cpp'
    exe_file=a+'.exe'
    os.system('g++  ' + inp_file + '  -o  ' + exe_file)
    os.system(exe_file)
if __name__=='__main__':
        main(sys.argv[1:])
```

மேற்கண்ட நிரலின் வெளியீடு

```
C:\Users\Dell>python c:\pyprg\main.py -i c:\pyprg\ch14\hello
c:\pyprg\hello.cpp: In function 'int main()':
c:\pyprg\hello.cpp:6:21: error: expected ';' before 'return'
  std::cout<<"hello"
                    ^
                    ;
  return 0;
  ~~~~~~
'c:\pyprg\hello.exe' is not recognized as an internal or external command,
operable program or batch file.
```

குறிப்பு

மேற்கண்ட நிரலில், பைத்தான் உள்ள பிழைகளை வெளிக்காட்ட உதவுகிறது. பிழைகள் அவற்றின் வரிசை எண்களுடன் வெளியிடப்படுகின்றன. அவற்றின் வரிசை எண், பைத்தான் script-லிருந்து தொடங்குகிறது.

Chapter 14.indd 285 10-02-2020 20:04:02

14.9 அணியைக் கொண்டுள்ள C++ நிரலை இயக்கும் பைத்தான் நிரல்

கொடுக்கப்பட்ட எண் பாலிண்ட்ரோம் எண்ணா அல்லது இல்லையா என்று கண்டறிந்த நம்முடைய முந்தைய நிரலில், நீங்கள் கட்டுப்பாட்டு கட்டமைப்புக் கூற்றுகளை நிங்கள் பயன்படுத்தியுள்ளீர்கள். இப்பொழுது நீங்கள் அணியைக் கொண்டுள்ள C++ நிரலை இயக்கப் போகிறீர்கள்.

எடுத்துக்காட்டு :- 14.9.1

ஓர் அணியின் இடமாற்று அணிக்கோவையை (இருபரிமாக அணி) பெற ஒரு C++ நிரல் எழுதுக. இப்பொழுது, நோட்பேடில் File → New என்று தேர்ந்தெடுக்கப்படும் C++ நிரலை உள்ளீடு செய்யவும்.

```
#include <iostream>
using namespace std;
int main()
{
int a[3][3], i, j;
    for(i=0; i<3; i++)
    {
    for(j=0; j<3; j++)
    {   cout<<"enter the value for array["<<i+1<<"]"<<"["<<j+1<<"] :";
        cin>>a[i][j];
    }
}
    system("cls");
cout<<"\n\nOriginal Array\n";
for(i=0; i<3; i++)    {
for(j=0; j<3; j++)
cout<<a[i][j]<<' ';
cout<<endl;   }
cout<<"\n\n The Transpose of Matrix\n";
 for (int i = 0; i < 3; i++)
 {
    for (int j = 0; j < 3; j++)
        cout<<a[j][i]<<' ';
cout<<endl ;
}
    return 0;
}
```

இந்த் கோப்பை **trans_cpp.cpp** என்று சேமிக்கவும். இப்பொழுது நோட்பேடில் **File → New** என்று தேர்ந்தெடுக்கும் பைத்தான் நிரலை உள்ளீடு செய்யவும்.

File ஐ **transpose.py** என்று சேமிக்கவும். இது **.cpp file**ஐ தொகுத்து, இயக்குவதற்கான நிரலாகும்.

 # Python tanspose.py -i trans_cpp

```
import sys, os, getopt
def main(argv):
    opts, args = getopt.getopt(argv, "i:")
    for o, a in opts:
        if o in "-i":
            run(a)
def run(a):
    inp_file=a+'.cpp'
    exe_file=a+'.exe'
if __name__=='__main__':
        main(sys.argv[1:])
```

மேற்கண்ட நிரலின் வெளியீடு

```
Original Array
    1 2 3
    4 5 6
    7 8 9
The Transpose of Matrix
    1 4 7
    2 5 8
    3 6 9
```

14.10 செயற்கூறுகளைக் கொண்ட C++ நிரல்களை இயக்கும் பைத்தான் நிரல்

இப்பொழுது நீங்கள் செயற்கூறுகள் மற்றும் செயற்கூறு அழைப்புகள் கொண்ட C++ நிரலை இயக்க பைத்தான் Scriptஐ சோதிக்க போகிறீர்கள்.

எடுத்துக்காட்டு 14.10.1 – பயனர் வரையறுத்த செயற்கூறினைக் கொண்டு ஒரு எண்ணின் கனத்தினை கண்டறியும் C++ நிரலை எழுதுக.

```
#include <iostream>
using namespace std;
// Function declaration
int cube(int num);
int main()
{
int num;
int c;
cout<<"Enter any number: "<<endl;
```


Chapter 14.indd 287 10-02-2020 20:04:02

```
cin>>num;
c = cube(num);
cout<<"Cube of " <<num<< " is "<<c;
return 0;
}
//Function to find cube of any number
int cube(int num)
{
	return (num * num * num);
}
// Save this file as cube_file.cpp
#Now select File→New in Notepad and type the Python program
# Save the File as fun.py
# Program that compiles and executes a .cpp file
 # Python fun.py -i c:\pyprg\cube_file.cpp
import sys, os, getopt
def main(argv):
   opts, args = getopt.getopt(argv, "i:")
   for o, a in opts:
      if o in "-i":
         run(a)
def run(a):
   inp_file=a+'.cpp'
   exe_file=a+'.exe'
   os.system('g++  ' + inp_file + '  -o  ' + exe_file)
   os.system(exe_file)
if __name__=='__main__':
      main(sys.argv[1:])
```

மேற்கண்ட நிரலின் வெளியீடு

```
Enter any number:
5
Cube of 5 is 125
```


Chapter 14.indd 288 10-02-2020 20:04:02

14.11 ஓர் இனக்குழுவின் மரபுரிமத்தை விளக்கும் பைத்தான் நிரல்

நீங்கள் இப்பொழுது C++-ன் இனக்குழு நிரலை செயல்படுத்த போகிறீர்கள். இந்த இனக்குழு நிரல் பலநிலை மரபுரிமத்தைக் கொண்டுள்ளது. மேலும், இது பைத்தான் Scriptஐ பயன்படுத்தி வெளியீட்டை தருகிறது.

எடுத்துக்காட்டு 14.11.1 - பலநிலை மரபுரிமத்தை செயல்படுத்தும் C++நிரல்

```
// C++ program to implement Multilevel Inheritance
//Now select File→New in Notepad and type the C++ program
#include <iostream>
using namespace std;

// base class
class Vehicle
{
        public:
        Vehicle()
        {
            cout<< "This is a Vehicle" <<endl;
        }
};
```

```
class threeWheeler: public Vehicle
{  public:
threeWheeler()
        {
            cout<<"Objects with 3 wheels are vehicles"<<endl;
        }
};
// sub class derived from two base classes
class Auto: public threeWheeler{
        public:
            Auto()
        {
            cout<<"Auto has 3 Wheels"<<endl;
        }
};
// main function
int main()
{
        //creating object of sub class will invoke the constructor of base classes
        Auto obj;
        return 0;
}
```

```
// Save this file as inheri_cpp.cpp
//Now select File → New in Notepad and type the Python program
# Save the File as classpy.py
# Python classpy.py -i  inheri_cpp  command to execute c++ program
import sys, os, getopt
def main(argv):
    opts, args = getopt.getopt(argv, "i:")
    for o, a in opts:
        if o in "-i":
            run(a)
def run(a):
    inp_file=a+'.cpp'
    exe_file=a+'.exe'
    os.system('g++  ' + inp_file + '  -o  ' + exe_file)
    os.system(exe_file)
if __name__=='__main__':
        main(sys.argv[1:])
```

மேற்கண்ட நிரலின் வெளியீடு

```
This is a Vehicle
Objects with 3 wheels are vehicles
Auto has 3 Wheels
```

இந்த எடுத்துக்காட்டிலிருந்து, C++ நிரல்களை ஒருங்கிணைக்கப் பயன்படுத்தப்படும் பைத்தான் Script பொதுவானது என்பதை புரிந்துக் கொள்ளலாம். பைத்தான் Script கோப்பு மற்றும் C++ (cpp) கோப்பின் பெயர்கள் மட்டும் மாற்றியமைக்கப்பட்டுள்ளது. ஒருவருடைய கணினியில் முன்பே C++ கோப்பு இருந்தால், C++குறிமுறையை உள்ளீடு செய்வது கட்டாயமில்லை என்பதை கவனத்தில் கொள்ளவும். இந்தப் பாடப்பகுதியில், பைத்தான் மூலம் சாதாரண செயற்கூறுவைப் போல் மரபுரிமத்தை போன்ற அனைத்து வகை c++ நிரல்களிலும் செயல்படுத்த முடியும் என்பதைக் காண்பிக்க C++ நிரல் குறிமுறை வெளிப்படையாக குறிப்பிடப்பட்டுள்ளது. இத்தகைய பைத்தான் Script ஐ பயன்படுத்தி, நீங்கள் C நிரல்களையும் தொகுத்து, செயல்படுத்திக் கொள்ளலாம். பைத்தான், அதன் IDE இல்லாமலேயே தன்னிச்சையாக C++ நிரல் கோப்பை செயல்படுத்துவதால், அதனை Scripting மொழி எனவும் அழைக்கலாம்.

Chapter 14.indd 290
10-02-2020 20:04:02

நினைவில் கொள்க

- C++ ஒரு தொகுப்பு மொழி மாற்றி மொழி, பைத்தான் என்பது வரி மொழி மாற்றி மொழியாகும்.
- C++ நிலையான தன்மைக் கொண்ட தொகுக்கப்பட்ட மொழி, ஆனால் பைத்தான் என்பது மாறும் தன்மை கொண்ட மொழி.
- C++ போன்ற நிலையான வகையை சார்ந்த மொழியில் நிரலர் கணிப்பொறிக்கு ஒவ்வொரு தரவு மதிப்பும் பயன்படுத்தக்கூடிய தரவின வகையை வெளிப்படையாக தெரிவித்தல் அவசியமாகிறது.
- பைத்தான் போன்று மாறும் நிலைக் கொண்ட மொழிக்கு தரவு மதிப்பிற்கான தரவு வகையை வெளிப்படையாக கொடுக்கப்பட்ட வேண்டிய தேவையில்லை .
- பைத்தான் தரவு மதிப்பின் அடிப்படையில் மாறியை கையாள்கிறது.
- ஒரு Scripting மொழி என்பது பிற நிரலாக்க மொழிகளுடன் ஒருங்கிணைப்பதற்கும், தொடர்பு கொள்வதற்கும் வடிவமைக்கப்பட்ட ஒரு நிரலாக்க மொழியாகும்.
- MinGW இயக்க நேர தலைப்புக் கோப்புகளின் தொகுப்பைக் குறிக்கிறது. விண்டோஸ் இயக்க முறைமையில் இயங்கும் வகையில் C, C++ மற்றும் FORTRAN நிரல் குறிமுறைகளை தொகுக்கவும், தொடர்புறுத்தவும் இது பயன்படுகிறது.
- (.) புள்ளி செயற்குறி தருவிக்கப்பட்ட கூறுநிலையின் செயற்கூறுகளை அணுகுவதற்கு பயன்படுகிறது.
- sys கூறுநிலை வரிமொழி மாற்றியால் பயன்படுத்தப்படும் மாறிகளுக்கும், வரிமொழி மாற்றியுடன் வலுவாக ஊடாடும் செயற்கூறுகளுக்கும் அணுகுதலை வழங்குகிறது.
- பைத்தானில் OS கூறுநிலை இயக்க அமைப்பின் சார்பு நிலை செயல்பாட்டுடன் பயன்படுத்த வழிவகை வழங்குகிறது.
- பைத்தானில் getopt கூறுநிலை கட்டளைவரி தேர்வுகளையும் செயலுருபுகளையும் பிரித்தெடுக்க உதவுகிறது.

செய்முறைப் பயிற்சி

1. கீழ்க்காணும் வரையறுப்புகளுடன் student என்னும் ஓர் இனக்குழுவை வரையறுக்க ஒரு C++ நிரலை எழுதவும்.

    Protected உறுப்பு

    Rno integer

    Public members

    void Readno(int); to accept roll number and assign to Rno

    void Writeno(); To display Rno.

    The class Test is derived Publically from the Student class contains the following details

    Protected உறுப்புகள்

    Mark1 float

    Mark2 float

    Public உறுப்புகள்

Chapter 14.indd 291 10-02-2020 20:04:02

void Readmark(float, float); To accept mark1 and mark2

void Writemark(); To display the marks

கீழ்க்காணும் வரையறுப்புகளுடன் sports என்னும் இனக்குழுவை உருவாக்கவும்.

Protected உறுப்புகள்

score integer

Public உறுப்புகள்

void Readscore(int); To accept the score

void Writescore(); To display the score

test இனக்குழுவிலிருந்து public அணுகியல்புடன் தருவிக்கப்பட்ட Result என்னும் இனக்குழு கீழ்க்காணும் வரையறுப்புகளை கொண்டுள்ளது

Private உறுப்புகள்

Total float

Public உறுப்புகள்

void display() assign the sum of mark1, mark2, score in total.

invokeWriteno(), Writemark() and Writescore(). Display the total also.

Total ஐ திரையில் காட்டும்.

C++ நிரலினை hybrid என்ற பெயரில் சேமிக்கவும். Hybrid.cpp என்ற கோப்பை செயல்படுத்த ஒரு பைத்தான் நிரலை எழுதவும்.

2. Write a C++ program to print boundary elements of a matrix and name the file as Border.cpp. Write a python program to execute the Border.cpp

மதிப்பீடு

பகுதி–அ

சரியான விடையை தேர்ந்தெடுத்து எழுதுக (1 மதிப்பெண்)

1. பின்வருவனவற்றுள் எது Scripting மொழி அல்ல?

   (அ) ஜாவாஸ்கிரிப்ட் (ஆ) PHP (இ) பெர்ல் (ஈ) HTML

2. பைத்தான் நிரலில் C++ நிரலை தருவித்தல் எவ்வாறு அழைக்கப்படுகிறது?

   (அ) wrapping செய்தல் (ஆ) பதிவிறக்கம் செய்தல்

   (இ) இணைத்தல் (ஈ) பிரித்தல்

3. API ன் விரிவாக்கம் is

   (அ) Application Programming Interpreter

   (ஆ) Application Programming Interface

   (இ) Application Performing Interface

Chapter 14.indd 292 10-02-2020 20:04:02

(ஈ) Application Programming Interlink

4. பைத்தான் மற்றும் C++ நிரல்களை இடைமுகப்படுத்துவதற்கான கட்டமைப்பு

(அ) Ctypes (ஆ) SWIG (இ) Cython (ஈ) Boost

5. பின்வருவனவற்றுள் எது உங்கள் குறிமுறையை தனித்தனி பகுதிகளாக பிரித்தெடுப்பதற்கான மென்பொருள் வடிவமைப்பு தொழில்நுட்பம்?

(அ) பொருள்நோக்கு நிரலாக்கம்

(ஆ) கூறுநிலை நிரலாக்கம்

(இ) குறைந்த நிலை மொழி நிரலாக்கம்

(ஈ) செயல்முறை நோக்கு நிரலாக்கம்

6. நீங்கள் விண்டோஸ் இயக்க முறைமையுடன் தொடர்பு கொள்ள எந்த கூறுநிலை அனுமதிக்கிறது?

(அ) OS கூறுநிலை (ஆ) sys கூறுநிலை (இ) csv கூறுநிலை (ஈ) getopt கூறுநிலை

7. சரங்களை எந்த மாதிரியாக பிரிக்கும்பொழுது பிழையின்றி அமைந்தால், getopt( ) வெற்று அணியை திருப்பி அனுப்பும்?

(அ) argv மாறி (ஆ) opt மாறி (இ) args மாறி (ஈ) ifile மாறி

8. பின்வரும் நிரல் பகுதியில் உள்ள செயற்கூறின் பெயரை அடையாளம் காண்க.

```
if __name__ =='__main__':
        main(sys.argv[1:])
```

(அ) main(sys.argv[1:]) (ஆ) __name__ (இ) __main__ (ஈ) argv

9. கீழ்க்கண்டவற்றுள் எது உரை, எண்கள், படங்கள் மற்றும் அறிவியல் சார்ந்த தரவுகளை செயலாக்கப் பயன்படும்?

(அ) HTML (ஆ) C (இ) C++ (ஈ) PYTHON

10. __name__ இது எதனை கொண்டுள்ளது ?

(அ) c++ filename (ஆ) main() name

(இ) python filename (ஈ) os module name

பகுதி–ஆ

அனைத்து வினாக்களுக்கும் விடையளி (2 மதிப்பெண்)

1. Scripting மொழிக்கும், மற்ற நிரலாக்க மொழிக்கும் உள்ள தத்துவர்த்த வேறுபாடு யாது?
2. தொகுப்பான் மற்றும் வரிமொழி மாற்றியை வேறுபடுத்துக.
3. விரிவாக்கம் தருக (i) SWIG (ii) MinGW
4. கூறுநிலைகளின் பயன் யாது?
5. cd கட்டளையின் பயன் யாது? எடுத்துக்காட்டு தருக.

Chapter 14.indd 293 10-02-2020 20:04:02

பகுதி–இ

அனைத்து வினாக்களுக்கும் விடையளி **(3 மதிப்பெண்)**

1. பைத்தான் மற்றும் C++ வேறுபடுத்துக
2. Scripting மொழியின் பயன்பாடுகள் யாவை?
3. MinGW என்றால் என்ன? அதன் பயன் யாது?
4. கீழ்க்காணும் கூற்றில் கூறுநிலை, செயற்குறி, வரையிறையின் பெயர் ஆகியவற்றை அடையாளம் காண்க.

   welcome.display()
5. sys.argv என்றால் என்ன?

பகுதி ஈ

அனைத்து வினாக்களுக்கும் விடையளி **(5 மதிப்பெண்)**

1 பைத்தானில் ஏதேனும் 5 பண்புக்கூறுகளை கூறவும்.

2. பின்வரும் கட்டளை ஒவ்வொன்றையும் விளக்கவும்.

   Python <filename.py> -<i> <C++ filename without cpp extension>

3. பைத்தானில், sys,os,getopt கூறுநிலைகளின் தேவை என்ன என்பதை விளக்குக.

4. getopt() என்ற செயற்கூறின் தொடரியலை எழுதி, அதன் செயலுருபுகளையும், திருப்பியனுப்பும் மதிப்புகளையும் விளக்குக.

5. கீழ்க்காணும் c++ நிரலை செயல்படுத்த ஒரு பைத்தான் நிரலை எழுதவும்.

```
#include <iostream>
using namespace std;
int main()
{ cout<<"WELCOME";
return(0);
}
```

The above C++ program is saved in a file welcome.cpp

REFERENCES

1. *Learn Python The Hard Way by Zed Shaw*
2. *Python Programming Advanced by Adam Stuart or Powerful Python by Aaron Maxwell*
3. *https://docs.python.org*

Chapter 14.indd 294 10-02-2020 20:04:02