Unit V

CHAPTER 14

IMPORTING C++ PROGRAMS IN PYTHON

Learning Objectives

After the completion of this chapter, the student will be able to

- Understand what is wrapping
- Able to import C++ functions and classes in to Python programs
- Create environment to work with both languages
- Execute and debug Python programs

14.1 Introduction

Python and C++ are general-purpose programming language. However, Python is quite different from C++.

S.NO	PYTHON	C++
1	Python is typically an "interpreted" language	C++ is typically a "compiled" language
2	Python is a dynamic-typed language	C++ is compiled statically typed language
3	Data type is not required while declaring variable	Data type is required while declaring variable
4	It can act both as scripting and general purpose language	It is a general purpose language

Yet these two languages complement one another perfectly. Python is mostly used as a scripting or "glue", language. That is, the top level program mostly calls routines written in C or C++. This is useful when the logic can be written in terms of existing code (For example a program written in C++) but can be called and manipulated through Python program.

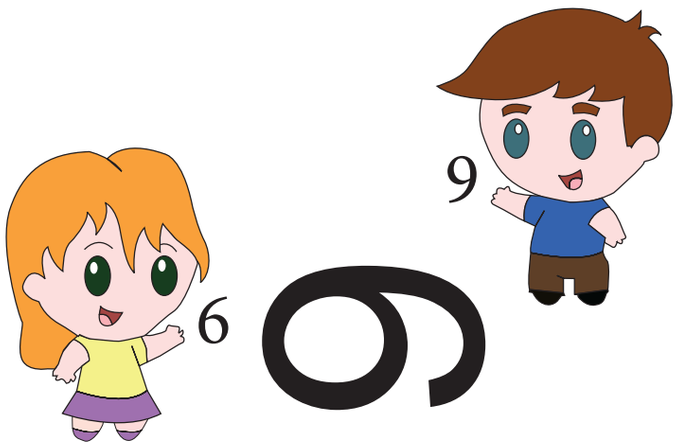
12th Computer Science_EM Chapter 14.indd 267 22-12-2021 16:38:25

14.2 Scripting Language

A scripting language is a programming language designed for integrating and communicating with other programming languages. Some of the most widely used scripting languages are JavaScript, VBScript, PHP, Perl, Python, Ruby, ASP and Tcl. Since a scripting language is normally used in conjunction with another programming language, they are often found alongside HTML, Java or C++.

14.2.1 Difference between Scripting and Programming Languages

Scripting Language and Programming Language looks like the following picture.



Basically, all scripting languages are programming languages. The theoretical difference between the two is that scripting languages do not require the compilation step and are rather interpreted. For example, normally, a C++ program needs to be compiled before running whereas, a scripting language like JavaScript or Python need not be compiled. A scripting language requires an interpreter while a programming language requires a compiler. A given language can be called as a scripting or programming language depending on the environment they are put to use.

14.3 Applications of Scripting Languages

1. To automate certain tasks in a program
2. Extracting information from a data set
3. Less code intensive as compared to traditional programming language
4. can bring new functions to applications and glue complex systems together

Python is actually an interpreted, high-level, general-purpose programming language that can be used on any modern computer operating system. It can be used for processing text, numbers, images, scientific data and just about anything else you might save on a computer. Now a days, large applications are written almost exclusively in Python.

12th Computer Science_EM Chapter 14.indd 268 22-12-2021 16:38:25

14.4 Features of Python over C++

- Python uses Automatic Garbage Collection whereas C++ does not.
- C++ is a statically typed language, while Python is a dynamically typed language.
- Python runs through an interpreter, while C++ is pre-compiled.
- Python code tends to be 5 to 10 times shorter than that written in C++.
- In Python, there is no need to declare types explicitly where as it should be done in C++
- In Python, a function may accept an argument of any type, and return multiple values without any kind of declaration beforehand. Whereas in C++ return statement can return only one value.

Note

Python deletes unwanted objects (built-in types or class instances) automatically to free the memory space. The process by which Python periodically frees and reclaims blocks of memory that no longer are in use is called Garbage Collection.

14.5 Importing C++ Files in Pythona

Importing C++ program in a Python program is called wrapping up of C++ in Python. Wrapping or creating Python interfaces for C++ programs are done in many ways. The commonly used interfaces are

- Python-C-API (API-**Application Programming Interface** for interfacing with C programs)
- Ctypes (for interfacing with c programs)
- SWIG (Simplified Wrapper Interface Generator- Both C and C++)
- Cython (Cython is both a Python-like language for writing C-extensions)
- Boost. Python (a framework for interfacing Python and C++)
- MinGW (*Minimalist GNU for Windows*)

14.5.1 MinGW Interface

MinGW refers to a set of runtime header files, used in compiling and linking the code of C, C++ and FORTRAN to be run on Windows Operating System.

MinGw-W64 (version of MinGW) is the best compiler for C++ on Windows. To compile and execute the C++ program, you need 'g++' for Windows. MinGW allows to compile and execute C++ program dynamically through Python program using g++.

Python program that contains the C++ coding can be executed through either by using command prompt or by using run terminal.

> g++ is a program that calls GCC (GNU C Compiler) and automatically links the required C++ library files to the object code.

Refer installation of MinGW in Annexure -2

14.5.2 Executing C++ Program through Python

1. Double click on the command prompt or the run terminal.

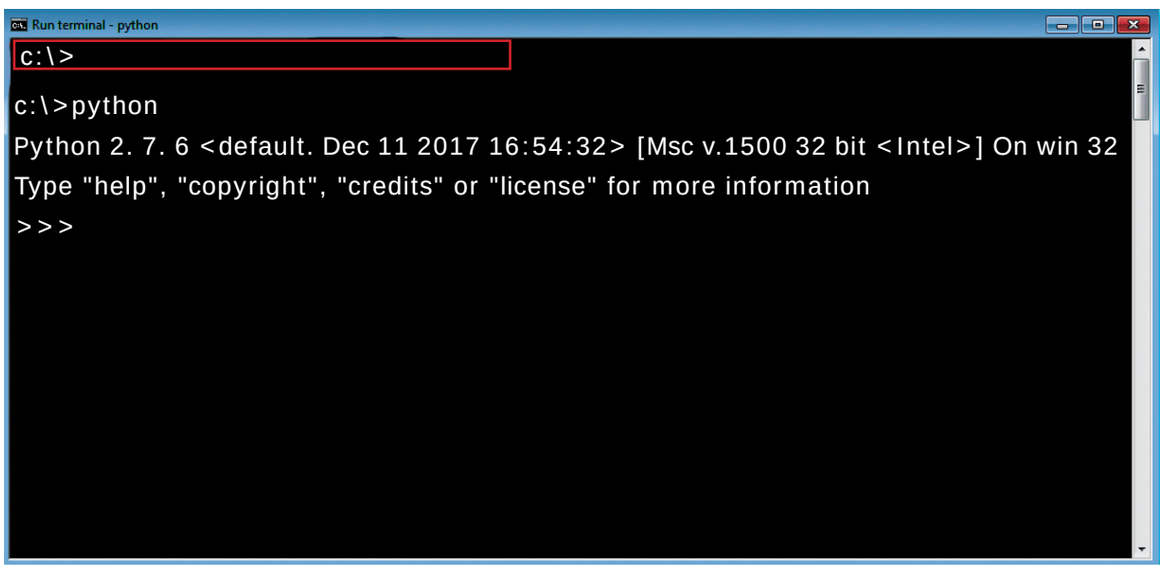


Figure 14.1

In this Example the prompt shows the "C:\>". See that highlighted area in the above window. To change a directory 'cd' command is used. For example to goto the directory pyprg, type the command 'cd pyprg' in the command prompt.

Consider the Example pycpp.py is a Python program which will read the C++program Pali.cpp. The "Pali.cpp" program accepts a number and display whether it is a "Palindrome or Not". For example the entered input number is 232 the output displayed will be "Palindrome". The C++ program Pali is typed in notepad and saved as **pali.cpp.** Same way the Python program **pycpp.py** code is also typed in notepad and saved as pycpp.py.

3. To execute our program double click the run terminal change the path to the Python folder location. The syntax to execute the Python program is

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

12th Computer Science_EM Chapter 14.indd 270 22-12-2021 16:38:25

Where,

Python	keyword to execute the Python program from command-line
filename.py	Name of the Python program to executed
- i	input mode
C++ filename without cpp extension	name of C++ file to be compiled and executed

For example type Python pycpp.py –i pali in the command prompt and press enter key. If the compilation is successful you will get the desired output. Otherwise the error will be displayed.

Note

In the execution command, the input file doesn't require its extension. For example, it is enough to mention just the name "pali" instead of "pali.cpp".

Now let us will see the execution through our example pycpp.py and pali.cpp. These two programs are stored in the folder c:\pyprg. If the programs are not located in same folder then the complete path must be specified for the files during execution. The output is displayed below

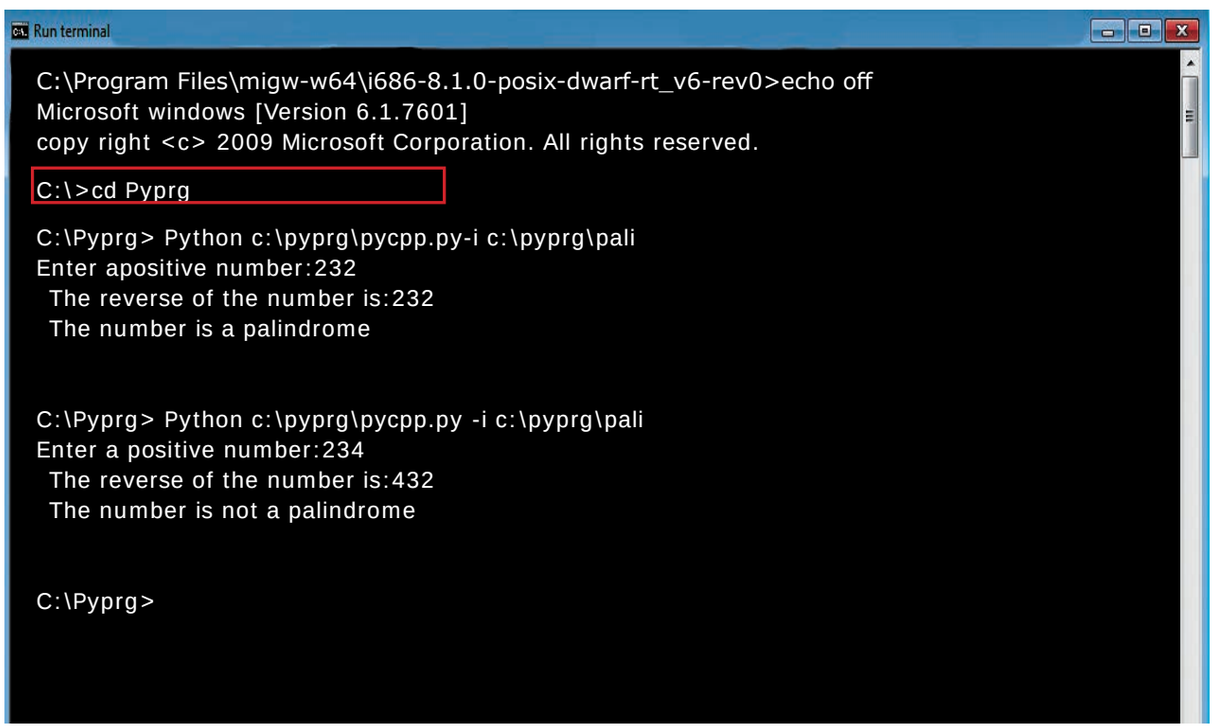


Fig 14.2

Note

To clear the screen in command window use **cls** command

Now let us will see how to write the Python program for compiling C++ code.

14.6 Python Program to import C++

Python contains many modules. For a problem Python allow programmers to have the flexibility in using different module as per their convenience. The Python program what we have written contains a few new commands which we have not come across in basic Python program. Since our program is an integration of two different languages, we have to import the modules like os, sys and getopt.

14.6.1 MODULE

Modular programming is a software design technique to split your code into separate parts. These parts are called modules. The focus for this separation should have modules with no or just few dependencies upon other modules. In other words: Minimization of dependencies is the goal.

But how do we create modules in Python? Modules refer to a file containing Python statements and definitions. A file containing Python code, for e.g. factorial.py, is called a module and its module name would be factorial. We use modules to break down large programs into small manageable and organized files. Furthermore, modules provide reusability of code. We can define our most used functions in a module and import it, instead of copying their definitions into different programs.

Example:

```
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)
```

Output:

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

The above example is named as factorial.py

12th Computer Science_EM Chapter 14.indd 272 22-12-2021 16:38:25

14.6.2 How to import modules in Python?

We can import the definitions inside a module to another module. We use the **import** keyword to do this. To import our previously defined module **factorial** we type the following in the Python prompt.

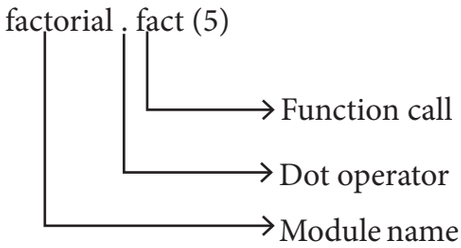
```
>>> import factorial
```

Using the module name we can access the functions defined inside the module. The dot (.) operator is used to access the functions. The syntax for accessing the functions from the module is

<module name> . <function name>

For example:

```
>>> factorial.fact(5)
120
```



Python has number of standard (built in) modules. Standard modules can be imported the same way as we import our user-defined modules. We are now going to see the Standard modules which are required for our program to run C++ code.

14.6.2.1 Python's sys module

This module provides access to builtin variables used by the interpreter. One among the variable in **sys** module is **argv**

sys.argv

sys.argv is the list of command-line arguments passed to the Python program. **argv contains** all the items that come via the command-line input, it's basically a list holding the command-line arguments of the program.

To use **sys.argv**, **import sys** should be used. The first argument, sys.argv[0] contains the name of the python program (example pali.py) and sys.argv [1]is the next argument passed to the program (here it is the C++ file), which will be the argument passed through main (). For example

12th Computer Science_EM Chapter 14.indd 273 22-12-2021 16:38:25

main(sys.argv[1])	The input file (C++ file) is send along with its path as a list(array) using **argv[1]**. **argv[0]** contains the Python program which need not be passed because by default __main__ contains source code reference.

14.6.2.2 Python's OS Module

The ***OS*** module in Python provides a way of using operating system dependent functionality.

The functions that the ***OS*** module allows you to interface with the Windows operating system where Python is running on.

os.system(): Execute the C++ compiling command (a string contains Unix, C command which also supports C++ command) in the shell (Here it is Command Window). For Example to compile C++ program **g++** ***compiler*** should be invoked. To do so the following command is used.

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

where each argument contains

os.system :-	function system() defined in **os** module to interact with the operating system
g++ :-	General compiler to compile C++ program under Windows Operating system.
variable_name1:-	Name of the C++ file along with its path and with extension .cpp in string format
mode :-	To specify input or output mode. Here it is o prefixed with hyphen.
variable_name2 :-	Name of the executable file with extension .exe in string format

For example the command to compile and execute C++ program is given below

os.system('g++ ' + cpp_file + ' -o ' + exe_file)	g++ compiler compiles the file cpp_file and –o (output) send to exe_file

Note

'+' in os.system() indicates that all strings are concatenated as a single string Therfore give a space after each word for the above argument. For example **'g++ ' + cpp_file + ' -o ' + exe_file**

12th Computer Science_EM Chapter 14.indd 274 22-12-2021 16:38:25

14.6.2.3.3 Python getopt module

The getopt module of Python helps you to parse (split) command-line options and arguments. This module provides getopt() method to enable command-line argument parsing.

getopt.getopt function

This function parses command-line options and parameter list. Following is the syntax for this method –

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

Here is the detail of the parameters –

argv – This is the argument list of values to be parsed (splited). In our program the complete command will be passed as a list. For example ***c:\pyprg\pali.py -i c:\pyprg\pali_cpp***

options – This is string of option letters that the Python program recognize as, for input or for output, with options (like 'i' or 'o') that followed by a colon (:). Here colon is used to denote the mode.

long_options – This contains a list of strings. Argument of Long options should be followed by an equal sign ('='). In our program the C++ file name along with its path will be passed as string and 'i' i will be also passed to indicate it as the input file.

getopt() method returns value consisting of two elements. Each of these values are stored separately in two different list (arrays) **opts** and **args** .Opts contains list of splitted strings like mode and path. **args** contains error string, if at all the comment is given with wrong path or mode. **args** will be an empty list if there is no error.

For example The Python code which is going to execute the **C++ file p4** in command line will have the getopt() method like the following one.

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

where **opts** contains	[('-i', 'c:\\pyprg\\p4')]
-i :-	**option - mode should be followed by : (colon)**
'c:\\pyprg\\p4'	**value - absolute path of C++ file.**

In our examples since the entire command line commands are parsed and no leftover argument, the **second argument args** will be empty []. If args is displayed using print() command it displays the output as [].

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


12th Computer Science_EM Chapter 14.indd 275 22-12-2021 16:38:25

Note

You can check out the full list of Python standard modules and what they are for. These files are in the Lib directory inside the location where Python is installed.

Some more command for wrapping C++ code

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

__name__ (A Special variable) in Python

Since there is no main() function in Python, when the command to run a Python program is given to the interpreter, the code that is at level 0 indentation (top must line of the program) is to be executed. However, before doing that, interpreter will define a few special variables. **__name__ is one such special variable which by default stores the name of the Python file.** If the source file is executed as the main program, the interpreter sets the __name__ variable to have a value "__main__".

__name__ is a built-in variable which evaluates to the name of the current module. Thus it can be used to check whether the current script is being run on its own.

For example consider the following

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

If the command line Python program itself is going to execute first, then __name__ contains the string " __main__". The condition if " __main__"==" __main__": is true then the main function is called.

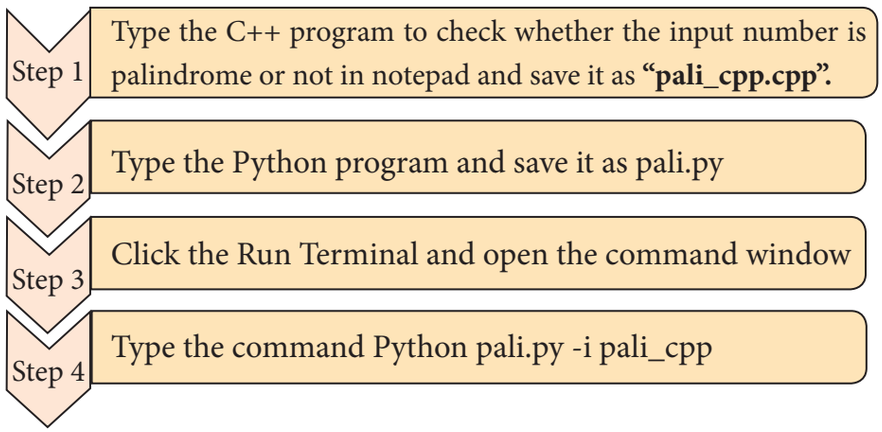
Note

sys.argv[1:] - get everything after the script name(file name).
sys.argv[0] is the script name (python program)
Remember "string slicing" you have studied in chapter 8.

14.7 Python program Executing C++ Program using control statement

Now let us write a Python program to read a C++ coding and execute its result. The steps for executing the C++ program to check a given number is palindrome or not is given below

12th Computer Science_EM Chapter 14.indd 276 22-12-2021 16:38:25



Example:- 14.7.1 - Write a C++ program to enter any number and check whether the number is palindrome or not using while loop.

```
/*. 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
```


12th Computer Science_EM Chapter 14.indd 277 22-12-2021 16:38:25

```
#Now select File→New in Notepad and type the Python program
# Save the File as pali.py . Program that compiles and executes a .cpp file
# 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\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
```


12th Computer Science_EM Chapter 14.indd 278 22-12-2021 16:38:25

Python code	How does it works
import sys, os, getopt	include sys , os and getopt modules to use the required function
def main(argv):	Function main() is defined and 'argv' contains the 'input mode and the c++ program file' in the form of list i.e ['-i', 'c:\ pyprg \pali_cpp']
opts, args = getopt.getopt(argv, "i:")	getopt() splits the command as option and argument. 'opts' contains [('-i', 'c:\pyprg\pali_cpp')]. Since no error 'args' shows []
for o, a in opts:	'o' contains the mode and 'a' contains the path of c++ program i.e print("o = ",o) shows o = -i print("a = ",a) shows a = c:\pyprg\pali_cpp
if o in ("-i"):	Checks o == 'i' if true
run(a)	Calls the function run() passed along with the c++ program
def run(a):	Definition of run() function begins here
inp_file=a+'.cpp'	Variable 'inp_file' contains the joined c++ program name and .cpp print(inp_file) shows c:\pyprg\pali_cpp.cpp
exe_file=a+'.exe'	Variable 'exe_file' contains the joined c++ program name and .exe print(exe_file) shows c:\pyprg\pali_cpp.exe
os.system('g++ ' + inp_file + ' -o ' + exe_file)	g++ compiler compiles the c++ program in inp_file and store the executable file in exe_file
os.system(exe_file)	Executes the exe file
if __name__=='__main__':	__name__ stores name of the python program __ main__ also stores the name of the python program.
main(sys.argv[1:])	If 'name' and 'main' are equal then main() is called and passed with the command line argument omitting the python program name argv[1:] contains –i c:\pyprg\pali_cpp

12th Computer Science_EM Chapter 14.indd 279 22-12-2021 16:38:25

The Python script(program) is mainly used to read the C++ file along with the type of mode like 'i'/'o'. 'getopt()' Parses(splits) each value of the command line and passes the options(values) as list to 'opt' and since no error 'args' generates empty list[]. Using 'for loop' the tuple in the list is unpacked - 'o' stores the mode and 'a' stores the name along with the path of the c++ file.

The variable 'inp_file' store the c++ file along with its extension and 'exe_file' stores the executable file with .exe extension. '+' usd in this program helps to concatenate the file name with the extensions. 'os.system()' along with 'g++' compiles the inp_file. Mode 'o' sends the executable file to 'exe_file'.

'__name__' variable directs the program to start from the beginning of the Python script(zero'th line) The "main()" definition does the Parsing and calling the run(). The "run()" invoke the "g++" compiler and creates the exe file. The system() of "os" module executes the .exe file and the desired output will be displayed on the output screen. The file extensions are added by the Python script so it is even possible to execute C programs.

14.8 How Python is handling the errors in C++

Python not only execute the successful C++ program, it also helps to display even errors if any in C++ statement during compilation. For example in the following C++ program an error is there. Let us see what happens when you compile through Python.

Example 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
```

```
# 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)
```


12th Computer Science_EM Chapter 14.indd 280 22-12-2021 16:38:25

```
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

```
C:\Users\Dell>python c:\pyprg\main.py -i c:\pyprg\hello
c:\pyprg\hello.cpp: In function 'int main()':
c:\pyprg\hello.cpp:7:19: 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.
```

Note

In the above program Python helps to display the error in C++. The error is displayed along with its line number. The line number starts from the beginning of the C++ program

14.9 Python program Executing C++ Program Containing Arrays

In our previous program to check whether a number is a palindrome number or not you have used only control structure statements. Now you are going to execute a C++ program containing array.

Example :- 14.9.1

Write a C++ program to print Transpose of a matrix(2 D array).*/

```
//Now select File->New in Notepad and type the  C++  program
#include <iostream>
using namespace std;
int main()
{
int a[3][3], i, j;
for(i=0; i<3; i++)
```


12th Computer Science_EM Chapter 14.indd 281 22-12-2021 16:38:26

```
{
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;
}
```

// Save this file as trans_cpp.cpp

//Now select File→New in Notepad and type the Python program

 # Save the File as transpose.py.Program that compiles and executes a .cpp file

 # 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:])
```


12th Computer Science_EM Chapter 14.indd 282 22-12-2021 16:38:26

Output of the above program

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

You would have noticed the Python program used to execute the transpose of a matrix is also the same as what you have used in palindrome program. From this what you have understood?

14.10 Python program Executing C++ Program Containing Functions

Now you are going to test the Python script to run a C++ program having functions and function calls.

Example 14.10.1 - Write a C++ program using a user defined function to function cube of a number

```
/*Write a C++ program using a user defined function to function  cube of a number.*/
//Now select File→New in Notepad and type the C++ program
#include <iostream>
using namespace std;
// Function declaration
int cube(int num);
int main()
{
int num;
int c;
cout<<"Enter any number: "<<endl;
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);
}
```


12th Computer Science_EM Chapter 14.indd 283 22-12-2021 16:38:26

```
// 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
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

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

14.11 Python program to Illustrate the inheritance of a Class

Now you are going to execute a class program of C++. The class program is of multilevel inheritance. This also gives output using Python script

Example 14.11.1 - C++ program to implement Multilevel Inheritance

```
// 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;
      }
};
```


12th Computer Science_EM Chapter 14.indd 284 22-12-2021 16:38:26

```
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:])
```

Output of the above program

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

From all these example One can understand the Python script(program) used for integrating C++ is common and very simple. Only the name of the Python script file and the C++ (cpp) file have changed. Remember it is not mandatory to type the C++ coding if it already exits. The various types of C++ programs like normal, function, inheritance program can be executed through python. Using the same Python script you can even compile and execute C program. Since python automates the C++ program file to execute without it's IDE, Python can be called as a Scripting Language.

Points to remember:

- C++ is a compiler based language while Python is an interpreter based language.
- C++is compiled statically whereas Python is interpreted dynamically
- A static typed language like C++ requires the programmer to explicitly tell the computer what "data type" each data value is going to use.
- A dynamic typed language like Python, doesn't require the data type to be given explicitly for the data. Python manipulate the variable based on the type of value.
- A scripting language is a programming language designed for integrating and communicating with other programming languages
- MinGW refers to a set of runtime header files, used in compiling and linking the code of C, C++ and FORTRAN to be run on Windows Operating System
- The dot (.) operator is used to access the functions of a imported module
- sys module provides access to some variables used by the interpreter and to functions that interact with the interpreter
- OS module in Python provides a way of using operating system dependent functionality
- The getopt module of Python helps you to parse (split) command-line options and arguments

12th Computer Science_EM Chapter 14.indd 286 22-12-2021 16:38:26

Hands on Experience

1. Write a C++ program to create a class called Student with the following details

Protected member

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 member

Mark1 float

Mark2 float

Public members

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

void Writemark(); To display the marks

Create a class called Sports with the following detail

Protected members

score integer

Public members

void Readscore(int); To accept the score

void Writescore(); To display the score

The class Result is derived Publically from Test and Sports class contains the following details

Private member

Total float

Public member

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

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

Save the C++ program in a file called hybrid. Write a python program to execute the hybrid.cpp

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

12th Computer Science_EM Chapter 14.indd 287 22-12-2021 16:38:26

Evaluation

Part - I

1. Which of the following is not a scripting language?

   (A) JavaScript (B) PHP

   (C) Perl (D) HTML

2. Importing C++ program in a Python program is called

   (A) wrapping (B) Downloading

   (C) Interconnecting (D) Parsing

3. The expansion of API is

   (A) Application Programming Interpreter

   (B) Application Programming Interface

   (C) Application Performing Interface

   (D) Application Programming Interlink

4. A framework for interfacing Python and C++ is

   (A) Ctypes (B) SWIG

   (C) Cython (D) Boost

5. Which of the following is a software design technique to split your code into separate parts?

   (A) Object oriented Programming

   (B) Modular programming

   (C) Low Level Programming

   (D) Procedure oriented Programming

6. The module which allows you to interface with the Windows operating system is

   (A) OS module (B) sys module

   (c) csv module (d) getopt module

7. getopt() will return an empty array if there is no error in splitting strings to

   (A) argv variable (B) opt variable

   (c)args variable (d) ifile variable

12th Computer Science_EM Chapter 14.indd 288 22-12-2021 16:38:26

8. Identify the function call statement in the following snippet.

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

(A) main(sys.argv[1:]) (B) __name__

(C) __main__ (D) argv

9. Which of the following can be used for processing text, numbers, images, and scientific data?

(A) HTML (B) C

(C) C++ (D) PYTHON

10. What does __name__ contains ?

(A) c++ filename (B) main() name

(C) python filename (D) os module name

Part - II

1. What is the theoretical difference between Scripting language and other programming language?
2. Differentiate compiler and interpreter.
3. Write the expansion of (i) SWIG (ii) MinGW
4. What is the use of modules?
5. What is the use of cd command. Give an example.

Part - III

1. Differentiate PYTHON and C++
2. What are the applications of scripting language?
3. What is MinGW? What is its use?
4. Identify the module ,operator, definition name for the following

   welcome.display()
5. What is sys.argv? What does it contain?

Part - IV

1 Write any 5 features of Python.

2. Explain each word of the following command.

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

12th Computer Science_EM Chapter 14.indd 289 22-12-2021 16:38:26

3. What is the purpose of sys,os,getopt module in Python.Explain

4. Write the syntax for getopt() and explain its arguments and return values

5. Write a Python program to execute the following c++ coding

```
#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*

12th Computer Science_EM Chapter 14.indd 290 22-12-2021 16:38:26