Chapter 14
Importing C++ Programs in Python

PART – I
I. Choose The Best Answer

Question 1.
Which of the following is not a scripting language?
(a) JavaScript
(b) PHP
(c) Perl
(d) HTML
Answer:
(d) HTML

Question 2.
Importing C++ program in a Python program is called ……………………….
(a) wrapping
(b) Downloading
(c) Interconnecting
(d) Parsing
Answer:
(a) wrapping

Question 3.
The expansion of API is ……………………..
(a) Application Programming Interpreter
(b) Application Programming Interface
(c) Application Performing Interface
(d) Application Programming Interlink
Answer:
(b) Application Programming Interface

Question 4.
A framework for interfacing Python and C++ is ………………………..
(a) Ctypes
(b) SWIG
(c) Cython
(d) Boost
Answer:
(d) Boost

Question 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
Answer:
(b) Modular programming

Question 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
Answer:
(a) OS module

Question 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
Answer:
(c) args variable

Question 8.
Identify the function call statement in the following snippet.
if_name =='_main_':
main(sys.argv[l:])
(a) main(sys.argvfl:])
(b) _name_
(c) _main_
(d) argv
Answer:
(b) _name_

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

(a) HTML
(b) C
(c) C++
(d) PYTHON
Answer:
(d) PYTHON

Question 10.
What does name contains?
(a) C++ filename
(b) main( ) name
(c) python filename
(d) os module name
Answer:
(c) python filename

PART – II
II. Answer The Following Questions

Question 1.
What is the theoretical difference between Scripting language and other programming language?
Answer:
The theoretical difference between the two is that scripting languages do not require the 228 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.

Question 2.
Differentiate compiler and interpreter?
Answer:
Compiler:

1. It converts the whole program at a time
2. It is faster
3. Error detection is difficult. Eg. C++

Interpreter:

1. line by line execution of the source code.
2. It is slow
3. It is easy Eg. Python

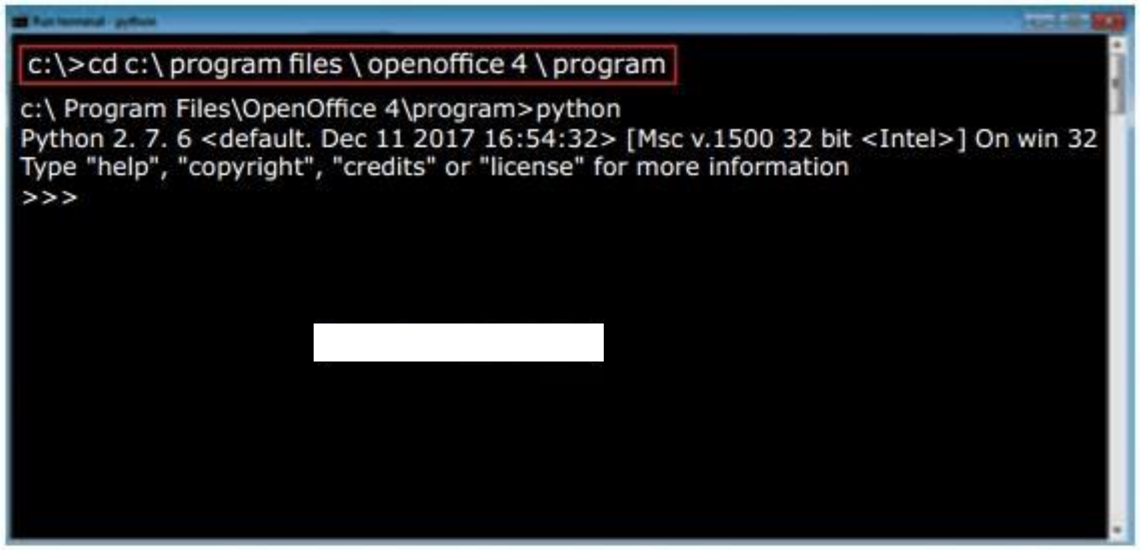
Question 3.
Write the expansion of

1. SWIG
2. MinGW

Answer:
SWIG (Simplified Wrapper Interface Generator. Both C and C++)
MinGW (Minimalist GNU for Windows)

Question 4.
What is the use of modules?
Answer:
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.

Question 5.
What is the use of cd command. Give an example?
Answer:
The syntax to change from c:\> to the folder where Python is located is
cd <absolute path>
where “cd” command refers to change directory and absolute path refers to the complete path where Python is installed.



```
c:\>cd c:\ program files \ openoffice 4 \ program
c:\ Program Files\OpenOffice 4\program>python
Python 2. 7. 6 <default. Dec 11 2017 16:54:32> [Msc v.1500 32 bit <Intel>] On win 32
Type "help", "copyright", "credits" or "license" for more information
>>>
```

[redacted]

In this Example to go to the folder where Python is located we should type the following command “cd C:\Program Files\OpenOffiice 4\Program”:

PART – III
III. Answer The Following Questions

Question 1.
Differentiate PYTHON and C++?
Answer:
PYTHON:

1. Python is typically an “interpreted” language
2. Python is a dynamic-typed language
3. Data type is not required while declaring variable
4. It can act both as scripting and general purpose language

C++:

1. C++ is typically a “compiled” language
2. C++ is compiled statically typed language
3. Data type is required while declaring variable
4. It is a general purpose language

Question 2.
What are the applications of scripting language?
Answer:
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

Question 3.
What is MinGW? What is its use?
Answer:
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 (versionofMinGW) 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 only through minGW-w64 project run terminal. The run terminal open the command-line window through which Python program should be executed.

Question 4.
Identify the module, operator, definition name for the following
welcome.display( )

Answer:
welcome – module name
– dot operator
display( ) – function name

Question 5.
What is sys.argv? What does it contain?
Answer:
sys.argv is the list of command-line arguments passed to the Python program, argv contains all the items that come along via the command-line input, it's basically an array holding the command-line arguments of the program.
To use sys.argv, you will first have to import sys. The first argument, sys.argv[0], is always the name of the program as it was invoked, and sys.argv[l] is the first argument you pass to the program (here it is the C++ file).
For example:
main(sys.args[1]) Accepts the program file (Python program) and the input file (C++ file) as a list(array). argv[0] contains the Python program which is need not to be passed because by default _main_ contains source code reference and argv[l] contains the name of the C++ file which is to be processed.

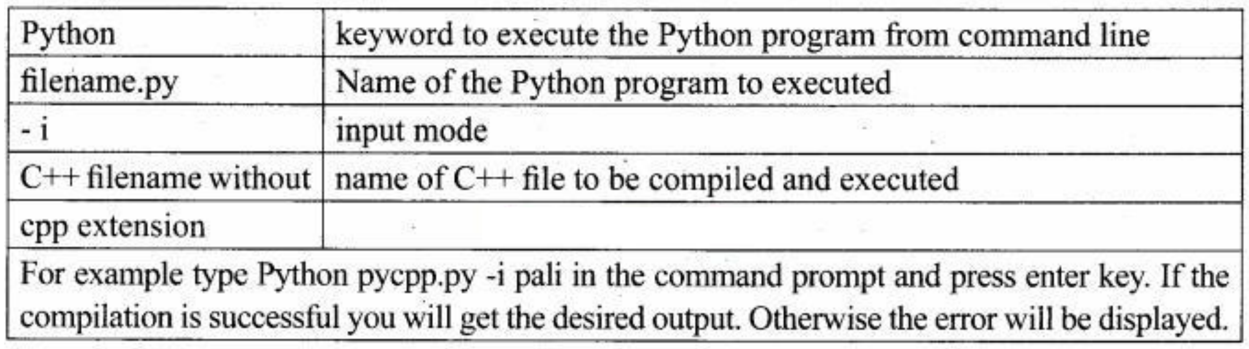
PART – IV
IV. Answer The Following Questions

Question 1.
Write any 5 features of Python?
Answer:

1. Python uses Automatic Garbage Collection
2. Python is a dynamically typed language.
3. Python runs through an interpreter.
4. Python code tends to be 5 to 10 times shorter than that written in C++.
5. In Python, there is no need to declare types explicitly
6. In Python, a function may accept an argument of any type, and return multiple values without any kind of declaration beforehand.

Question 2.
Explain each word of the following command?
Answer:
The syntax to execute the Python program is
Python <filename.py> -i <C++filename without cpp extension>

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	name of C++ file to be compiled and executed
cpp extension	
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.	

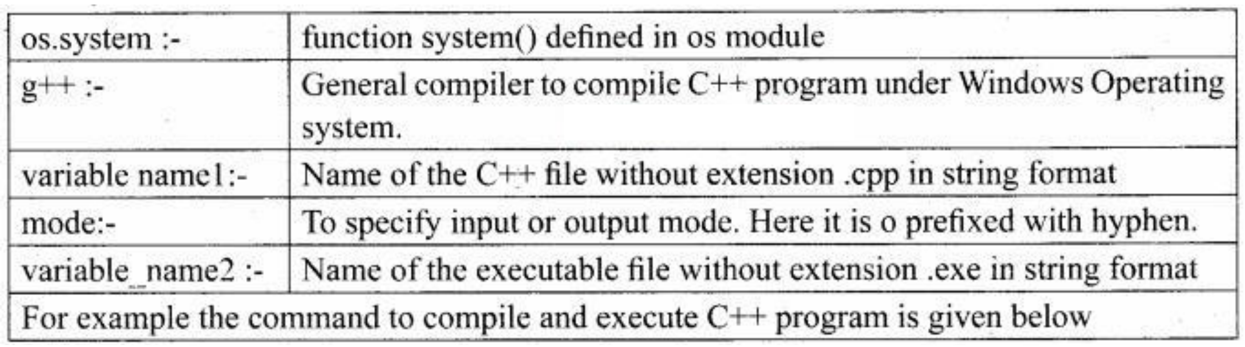
Question 3.
What is the purpose of sys,os,getopt module in Python. Explain?
Answer:
1. Python's sys module
This module provides access to some variables used by the interpreter and to functions that interact strongly with the interpreter.

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

To use sys.argv, you will first have to import sys. The first argument, sys.argv[0], is always the name of the program as it was invoked, and sys.argv) 1] is the first argument you pass to the program (here it is the C++ file). For example

main(sys.argv[1]) Accepts the program file (Python program) and the input file (C++ file) as a list(array). argv[0] contains the Python program which is need not to be passed because by default main contains source code reference and argv[l] contains the name of the C++ file which is to be processed.

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++' + <varciiable_namel> '-<mode>'+ <variable_name2>
where,



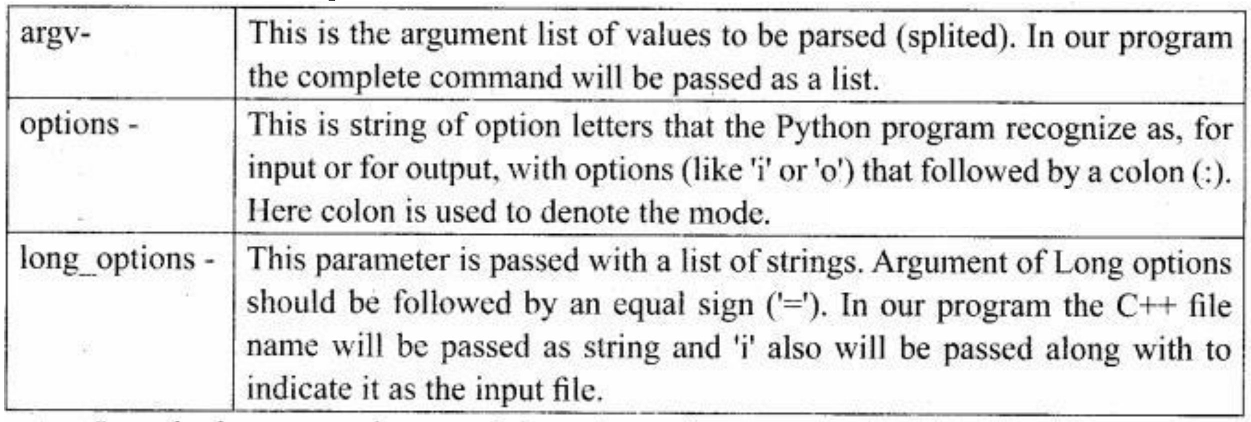
os.system :-	function system() defined in os module
g++ :-	General compiler to compile C++ program under Windows Operating system.
variable name1:-	Name of the C++ file without extension .cpp in string format
mode:-	To specify input or output mode. Here it is o prefixed with hyphen.
variable_name2 :-	Name of the executable file without 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 and send that as a List.

3. Python getopt module
The getopt module of Python helps you to parse (split) command-line options and arguments. This module provides two functions to enable command-line argument parsing,
getopt.getopt method
This method 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.
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 parameter is passed with a list of strings. Argument of Long options should be followed by an equal sign ('='). In our program the C++ fde name will be passed as string and 'i' also will be passed along with 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, path and args contains any string if at all not splitted because of wrong path or mode, args will be an empty array if there is no error in splitting strings by getopt( ).
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 opto contains [('-i', 'c:\ \pyprg\\p4')]
-i: – option nothing but mode should be followed by:
'c: \ \pyprg\\p4' value nothing but the 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 [ ].

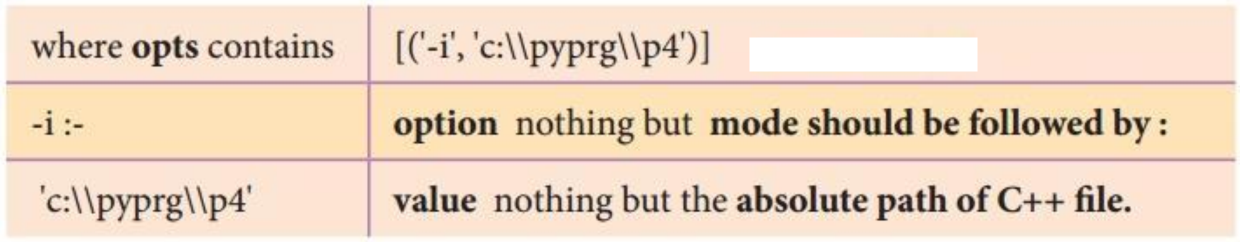
```
>>>print(args)
[ ]
```

Question 4.
Write the syntax for getopt( ) and explain its arguments and return values?
Answer:
Python getopt module
The getopt module of Python helps you to parse (split) command-line options and arguments. This module provides two functions to enable command-line argument parsing,
getopt.getopt method
This method parses command-line options and parameter list. Following is the syntax for this method –
<opts>,<args>=getopt.getopt(argx>, 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.
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 parameter is passed with a list of strings. Argument of Long options should be followed by an equal sign ('='). In our program the C++ file name will be passed as string and 'i' also will be passed along with to indicate it as the input file.

getopt( ) method returns value consisting of two elements. Each of these values are storec separately in two different list (arrays) opts and args .Opts contains list of splitted strings like mode, path and args contains any string if at all not splitted because of wrong path or mode, args will be an empty array if there is no error in splitting strings by getopt( ).
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** nothing but **mode should be followed by :**
'c:\\pyprg\\p4'	**value** nothing but the **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 display the output as [ ].
>>>print(args)
[ ]
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 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 file. If the source file is executed as the main program, the interpreter sets the _name_ variable to have a value as "_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 _main_ contains the name of that Python program and the Python special variable _name_ also contain the Python program name. If the condition is true it calls the main which is passed with C++ file as argument.

Question 5.
Write a Python program to execute the following C++ coding?
Answer:

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

The above C++ program is saved in a file welcome.cpp
Python program
Type in notepad and save as welcome.cpp

```
#include<iostream>
using namespace std;
int main( )
{
cout<<“WELCOME”;
retum(0);
}
```

Open Notepad and type python program and save as welcome.py

```
import sys,os,getopt
def main(argv):
cppfile =”
exefile = ”
opts, args = getopt.getopt(argv, “i:”, [ifile = ‘])
for o,a in opts:
if o in (“_i”,” ifile “):
cpp_file = a+ ‘.cpp’
exe_file = a+ ‘.exe’
run(cpp_file, exe_file)
def run(cpp_file, exe_file):
```

```
print(“compiling” +cpp_file)
os.system(‘g++’ +cpp_file + ‘_o’ + exe_file)
print(“Running” + exe_file)
print(“…………………………..”)
print
os.system(exe_file)
print
if — name — == ‘–main –‘;
main(sys.argv[1:])
```

Output:

```
———————-
WELCOME
———————-
```