Padasalai's Telegram Groups!

(தலைப்பிற்கு கீழே உள்ள லிங்கை கிளிக் செய்து குழுவில் இணையவும்!)

- Padasalai's NEWS - Group
https://t.me/joinchat/NIfCqVRBNj9hhV4wu6_NqA

- Padasalai's Channel - Group
https://t.me/padasalaichannel

- Lesson Plan - Group
https://t.me/joinchat/NIfCqVWwo5iL-21gpzrXLw

- 12th Standard - Group
https://t.me/Padasalai_12th

- 11th Standard - Group
https://t.me/Padasalai_11th

- 10th Standard - Group
https://t.me/Padasalai_10th

- 9th Standard - Group
https://t.me/Padasalai_9th

- 6th to 8th Standard - Group
https://t.me/Padasalai_6to8

- 1st to 5th Standard - Group
https://t.me/Padasalai_1to5

- TET - Group
https://t.me/Padasalai_TET

- PGTRB - Group
https://t.me/Padasalai_PGTRB

- TNPSC - Group
https://t.me/Padasalai_TNPSC

J. BASKARAN M.Sc., B.Ed. (C.S)
jbaskaran89@gmail.com
Puducherry.

J. ILAKKIA M.Sc., M.Phil., B.Ed. (C.S)
jilakkia@gmail.com
Puducherry.

COMPUTER SCIENCE

13. PYTHON AND CSV FILES

Section – A

Choose the best answer **(1 Mark)**

1. A CSV file is also known as a ….

(A) Flat File (B) 3D File (C) String File (D) Random File

2. The expansion of CRLF is

(A) Control Return and Line Feed (B) Carriage Return and Form Feed

(C) Control Router and Line Feed **(D) Carriage Return and Line Feed**

3. Which of the following module is provided by Python to do several operations on the CSV files?

(A) py (B) xls **(C) csv** (D) os

4. Which of the following mode is used when dealing with non-text files like image or exe files?

(A) Text mode **(B) Binary mode** (C) xls mode (D) csv mode

5. The command used to skip a row in a CSV file is

(A) next() (B) skip() (C) omit() (D) bounce()

6. Which of the following is a string used to terminate lines produced by writer()method of csv module?

(A) Line Terminator (B) Enter key (C) Form feed (D) Data Terminator

7. What is the output of the following program? import csv

```
d=csv.reader(open('c:\PYPRG\ch13\city.csv'))
next(d)
for row in d:
print(row)
```

if the file called "city.csv" contain the following details

```
chennai,mylapore
mumbai,andheri
```

A) chennai,mylapore **(B) mumbai,andheri**

(C) chennai (D) chennai,mylapore

mumba mumbai,andheri

8. Which of the following creates an object which maps data to a dictionary?

(A) listreader() (B) reader() (C) tuplereader() **(D) DictReader ()**

9. Making some changes in the data of the existing file or adding more data is called

(A)Editing (B) Appending **(C) Modification** (D) Alteration

10. What will be written inside the file test.csv using the following program import csv

```
D = [['Exam'],['Quarterly'],['Halfyearly']]
csv.register_dialect('M',lineterminator = '\n')
with open('c:\pyprg\ch13\line2.csv', 'w') as f:
wr = csv.writer(f,dialect='M')
wr.writerows(D)
f.close()
```

(A) Exam Quarterly Halfyearly

(B) Exam Quarterly Halfyearly

(C) E
Q
H

(D) Exam,
Quarterly,
Halfyearly

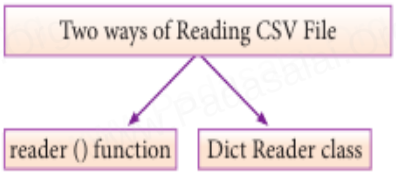
Section-B

Answer the following questions **(2 Marks)**

1. What is CSV File?

- A CSV file is a human readable text file where each line has a number of fields, separated by commas or some other delimiter.
- A CSV file is also known as a Flat File that can be imported to and exported from programs that store data in tables, such as ***Microsoft Excel*** or ***OpenOfficeCalc.***

2. Mention the two ways to read a CSV file using Python.



3. Mention the default modes of the File.

- The default is reading (‘r’) in text mode.
- In this mode, while reading from the file the data would be in the format of **strings.**

4. What is use of next() function?

- **“next()”command** is used to avoid or skip the first row or row heading.
- **Example:** While sorting the row heading is also get sorted, to avoid that the first is skipped using next().
- Then the list is sorted and displayed.

5. How will you sort more than one column from a csv file? Give an example statement.

- To sort by more than one column you can use **itemgetter** with multiple indices.

Syntax: **operator.itemgetter(col_no)**

Example: sortedlist = sorted (data, key=operator.itemgetter(1))

Section-C

Answer the following questions **(3 Marks)**

1. Write a note on open() function of python. What is the difference between the two methods?

- Python has a built-in function **open()** to open a file.
- This function returns a file object, also called a handle, as it is used to read or modify the file accordingly.
- The **default is reading** in text mode.
- In this mode, while reading from the file the data would be in the format of **strings.**
- On the other hand, binary mode returns bytes and this is the mode to be used when dealing with non-text files like image or exe files.

2. Write a Python program to modify an existing file.

- In this program, the third row of "student.csv" is modified and saved.
- First the "student.csv" file is read by using csv.reader() function.
- Then, the list() stores each row of the file.
- The statement "lines[3] = row", changed the third row of the file with the new content in "row".
- The file object writer using writerows (lines) writes the values of the list to "student.csv" file.

PROGRAM: student.csv

```
import csv
row = ['3', 'Meena','Bangalore']
with open('student.csv', 'r') as readFile:
reader = csv.reader(readFile)
lines = list(reader) # list()- to store each row of data as a list
lines[3] = row
with open('student.csv', 'w') as writeFile:
# returns the writer object which converts the user data with delimiter
writer = csv.writer(writeFile)
#writerows()method writes multiple rows to a csv file
writer.writerows(lines)
readFile.close()
writeFile.close()
```

3. Write a Python program to read a CSV file with default delimiter comma (,).

```
#importing csv
import csv                    #opening the csv file which is in different location with read mode
with open('c:\\pyprg\\sample1.csv', 'r') as F:
                              #other way to open the file is f= ('c:\\pyprg\\sample1.csv', 'r')
reader = csv.reader(F)        # printing each line of the Data row by row
print(row)
F.close()
```

OUTPUT:

```
['SNO', 'NAME', 'CITY']
['12101', 'RAM', 'CHENNAI']
['12102', 'LAVANYA', 'TIRUCHY']
['12103', 'LAKSHMAN', 'MADURAI']
```

4. What is the difference between the write mode and append mode.

Write Mode	Append Mode
• 'w'	• 'a'
• Open a file for writing.	• Open for appending at the end of the file without truncating it.
• Creates a new file if it does not exist or truncates the file if it exists.	• Creates a new file if it does not exist.

5. What is the difference between reader() and DictReader() function?

Reader():

- The reader function is designed to take each line of the file and make a list of all columns.
- Using this method one can read data from csv files of different formats like quotes (" "), pipe (|) and comma (,).
- csv. Reader work with list/tuple.
- **Syntax:** csv.reader(fileobject,delimiter,fmtparams)

DictReader():

- DictReader works by reading the first line of the CSV and using each comma separated value in this line as a dictionary key.
- DictReader is a class of csv module is used to read a CSV file into a dictionary.
- It creates an object which maps data to a dictionary.
- csv.DictReader work with dictionary.

Section - D

Answer the following questions: **(5 Marks)**

1. Differentiate Excel file and CSV file.

Excel	CSV
• Excel is a binary file that holds information about all the worksheets in a file, including both content and formatting.	• CSV format is a plain text format with a series of values separated by commas.
• XLS files can only be read by applications that have been especially written to read their format, and can only be written in the same way.	• CSV can be opened with any text editor in Windows like notepad, MS Excel, OpenOffice, etc.
• Excel is a spreadsheet that saves files into its own proprietary format viz. xls or xlsx	• CSV is a format for saving tabular information into a delimited text file with extension .csv
• Excel consumes more memory while importing data	• Importing CSV files can be much faster, and it also consumes less memory

2. Tabulate the different mode with its meaning.

Python File Modes:

Mode	Description
'r'	• Open a file for reading. (default)
'w'	• Open a file for writing. Creates a new file if it does not exist or truncates the file if it exists.
'x'	• Open a file for exclusive creation. If the file already exists, the operation fails.
'a'	• Open for appending at the end of the file without truncating it. Creates a new file if it does not exist.
't'	• Opren in text mode. (default)
'b'	• Open in binary mode.
'+'	• Open a file for updating (reading and writing)

3. Write the different methods to read a File in Python.

- Contents of CSV file can be read with the help of **csv.reader()** method.
- **The reader function is designed to take each line of the file and make a list of all columns.**
- Using this method one can read data from csv files of different formats like,
  1. CSV file - data with default delimiter comma (,)
  2. CSV file - data with Space at the beginning
  3. CSV file - data with quotes
  4. CSV file - data with custom Delimiters
- **The syntax for csv.reader() is** csv.reader(fileobject,delimiter,fmtparams**)**

i) CSV file with default delimiter comma (,)

The following program read a file called "sample1.csv" with default delimiter comma (,) and print row by row.

```
import csv
with open('c:\\pyprg\\sample1.csv', 'r') as F:
reader = csv.reader(F)
print(row)
F.close()
```

OUTPUT:

```
['SNO', 'NAME', 'CITY']
['12101', 'RAM', 'CHENNAI']
['12102', 'LAVANYA', 'TIRUCHY']
['12103', 'LAKSHMAN', 'MADURAI']
```

ii) CSV files- data with Spaces at the beginning

Consider the following file "sample2.csv" containing the following data when opened through notepad

Topic1,	Topic2,	Topic3,
one,	two,	three
Example1,	Example2,	Example3

The following program read the file through Python using "csv.reader()".

```
import csv
csv.register_dialect('myDialect',delimiter = ',',skipinitialspace=True)
F=open('c:\\pyprg\\sample2.csv','r')
reader = csv.reader(F, dialect='myDialect')
for row in reader:
print(row)
F.close()
```

OUTPUT:

```
['Topic1', 'Topic2', 'Topic3']
['one', 'two', 'three']
['Example1', 'Example2', 'Example3']
```

- These whitespaces in the data can be removed, by registering new dialects using **csv.register_dialect()** class of csv module.
- **A dialect describes the format of the csv file that is to be read.**
- In dialects the parameter **"skipinitialspace"** is used for removing whitespaces after the delimiter.

iii) CSV File-Data With Quotes

- You can read the csv file with quotes, by registering new dialects using csv.register_dialect() class of csv module.
- Here, we have quotes.csv file with following data.

SNO,Quotes

1, "The secret to getting ahead is getting started."

2, "Excellence is a continuous process and not an accident."

The following Program read "quotes.csv" file, where delimiter is comma (,) but the quotes are within quotes (" ").

```
import csv
csv.register_dialect('myDialect',delimiter = ',',quoting=csv.QUOTE_ALL,
skipinitialspace=True)
f=open('c:\\pyprg\\quotes.csv','r')
reader = csv.reader(f, dialect='myDialect')
for row in reader:
print(row)
```

OUTPUT:

```
['SNO', 'Quotes']
['1', 'The secret to getting ahead is getting started.']
['2', 'Excellence is a continuous process and not an accident.']
```

- In the above program, register a dialect with name myDialect.
- Then, we used **csv. QUOTE_ALL to display all the characters** after double quotes.

iv) CSV files with Custom Delimiters

- You can read CSV file having custom delimiter by registering a new dialect with the help of csv.register_dialect().

Roll No	Name	City
12101	Arun	Chennai
12102	Meena	Kovai
12103	Ram	Nellai
103	Ayush	M
104	Abinandh	M

- In the following file called "sample4.csv",each column is separated with | (Pipe symbol)

```
import csv
csv.register_dialect('myDialect', delimiter = '|')
with open('c:\\pyprg\\sample4.csv', 'r') as f:
    reader = csv.reader(f, dialect='myDialect')
    for row in reader:
        print(row)
f.close()
```

OUTPUT

```
['RollNo', 'Name', 'City']
['12101', 'Arun', 'Chennai']
['12102', 'Meena', 'Kovai']
['12103', 'Ram', 'Nellai']
```

4. Write a Python program to write a CSV File with custom quotes.

```
import csv
info = [[‘SNO’, ‘Person’, ‘DOB’],
[‘1’, ‘Madhu’, ‘18/12/2001’],
[‘2’, ‘Sowmya’,’19/2/1998’],
[‘3’, ‘Sangeetha’,’20/3/1999’],
[‘4’, ‘Eshwar’, ‘21/4/2000’],
[‘5’, ‘Anand’, ‘22/5/2001’]]
csv.register_dialect(‘myDialect’,quoting=csv.QUOTE_ALL)
with open(‘c:\\pyprg\\ch13\\person.csv’, ‘w’) as f:
writer = csv.writer(f, dialect=’myDialect’)
for row in info:
writer.writerow(row)
f.close()
```

OUTPUT :

“SNO”,”Person”,”DOB” ”1”,”Madhu”,”18/12/2001” ”2”,”Sowmya”,”19/2/1998”
”3”,”Sangeetha”,”20/3/1999” ”4”,”Eshwar”,”21/4/2000”
“5”,”Anand”,”22/5/2001”

5. Write the rules to be followed to format the data in a CSV file.

1. Each record (row of data) is to be located on a separate line, delimited by a line break by pressing enter key.

For example: ↵

xxx,yyy↵

↵ denotes enter Key to be pressed

2. The last record in the file may or may not have an ending line break.

For example:

ppp, qqq ↵
yyy, xxx

3.

- There may be an optional header line appearing as the first line of the file with the same format as normal record lines.
- The header will contain names corresponding to the fields in the file and should contain the same number of fields as the records in the rest of the file.
- **For example:** field_name1,field_name2,field_name3

```
aaa,bbb,ccc ↵
zzz,yyy,xxx CRLF( Carriage Return and Line Feed)
```

4.

- Within the header and each record, there may be one or more fields, separated by commas.
- Spaces are considered part of a field and should not be ignored.
- The last field in the record must not be followed by a comma.

For example: Red , Blue

5.

- Each field may or may not be enclosed in double quotes.
- If fields are not enclosed with double quotes, then double quotes may not appear inside the fields.

For example:

```
"Red","Blue","Green"↵     #Field data with doule quotes
Black,White,Yellow        #Field data without doule quotes
```

6.

- Fields containing line breaks (CRLF), double quotes, and commas should be enclosed in double-quotes.
- **For example:**

```
Red, ",", Blue CRLF   # comma itself is a field value.so it is enclosed with double quotes
Red, Blue , Green
```

7.

- If double-quotes are used to enclose fields, then a double-quote appearing inside a field must be preceded with another double quote.
- **For example:**

```
"Red, " "Blue", "Green",   # since double quotes is a field value it is enclosed with another double quotes
, , White
```

PREPARED BY

J. BASKARAN M.Sc., B.Ed. (C.S)
jbaskaran89@gmail.com

J. ILAKKIA M.Sc., M.Phil., B.Ed. (C.S)
jilakkia@gmail.com

J. BASKARAN M.Sc., B.Ed. (C.S)
jbaskaran89@gmail.com
Puducherry.

J. ILAKKIA M.Sc., M.Phil., B.Ed. (C.S)
jilakkia@gmail.com
Puducherry.

COMPUTER SCIENCE

14. IMPORTING C++ PROGRAMS IN PYTHON

Section – A

Choose the best answer **(1 Mark)**

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
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

Section-B

Answer the following questions **(2 Marks)**

1. What is the theoretical difference between Scripting language and other programming language?

Scripting Language	Programming Language
A scripting language requires an interpreter.	A programming language requires a compiler.
A scripting language need not be compiled.	A programming languages needs to be compiled before running .
Example: JavaScript, VBScript, PHP, Perl, Python, Ruby, ASP and Tcl.	**Example:** C, C++, Java, C# etc.

2. Differentiate compiler and interpreter.

Compiler	Interpreter
Compiler generates an Intermediate Code.	Interpreter generates Machine Code.
Compiler reads entire program for compilation.	Interpreter reads single statement at a time for interpretation.
Error deduction is difficult	Error deduction is easy
Comparatively faster	Slower
Example: gcc, g++, Borland TurboC	**Example:** Python, Basic, Java

3. Write the expansion of (i) SWIG (ii) MinGW

SWIG - Simplified Wrapper Interface Generator - Both C and C++

MinGW - Minimalist GNU for Windows

4. What is the use of modules?

- Modules are used to break down large programs into small manageable and organized files.
- 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.

5. What is the use of cd command. Give an example.

- **Syntax:** cd <absolute path>
- **"cd"** command used to change directory and absolute path refers to the complete path where Python is installed.
- **Example:** c:\>cd c:\ program files \ openoffice 4 \ program

Section-C

Answer the following questions **(3 Marks)**

1. Differentiate PYTHON and C++.

PYTHON	C++
• Python is typically an "interpreted" language	• C++ is typically a "compiled" language
• Python is a dynamic-typed language	• C++ is compiled statically typed language
• Data type is not required while declaring variable	• Data type is required while declaring variable
• It can act both as scripting and general purpose language	• It is a general purpose language

2. What are the applications of scripting language?

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

3. What is MinGW? What is its use?

- MinGW refers to a set of runtime header files.
- It is used in compiling and linking the code of C, C++ and FORTRAN to be run on Windows Operating System.
- MinGW allows to compile and execute C++ program dynamically through Python program using g++.

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

Welcome → Module name
. → Dot operator
display() → Function call

5. What is sys.argv? What does it contain?

- **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.
- **sys.argv[0]** is always the name of the program as it was invoked.

- **sys.argv[1]** is the first argument you pass to the program.
- **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
  - **argv[1]** contains the name of the C++ file which is to be processed.

Section - D

Answer the following questions: **(5 Marks)**

1. Write any 5 features of Python.

- Python uses Automatic Garbage Collection.
- Python is a dynamically typed language.
- Python runs through an interpreter.
- 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.
- In Python, a function may accept an argument of any type, and return multiple values without any kind of declaration beforehand.

2. Explain each word of the following command.

COMMAND: **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 cpp extension>	Name of C++ file to be compiled and executed

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

(i) 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** 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.
- **sys.argv[0]** is always the name of the program as it was invoked.
- **sys.argv[1]** is the first argument you pass to the program.
- **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
  - **argv[1]** contains the name of the C++ file which is to be processed.

(ii) 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 in the shell.
- For Example to compile C++ program g++ compiler should be invoked.
- **Command:** os.system ('g++' + <varaiable_name1> '-<mode>' + <variable_name2>

• **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

- **Example:**

  os.system('g++ ' + cpp_file + ' -o ' + exe_file) --

  g++ compiler compiles the file cpp_file and –o (output) send to exe_file

(iii) 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.

- **Syntax of getopt 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++ 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 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().
- **Example:**
- **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 displays the output as [].
- **Example:**
- >>>print(args)
- []

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

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.
- **Syntax of getopt 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++ 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 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().
- **Example:**
- **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 displays the output as [].
- **Example:**
- >>>print(args)
- []

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

C++ CODE:

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

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

PYTHON PROGRAM:

```
import sys, os, getopt
def main(argv):
  cpp_file = ''
  exe_file = ''
  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__':      #program starts executing from here
 main(sys.argv[1:])
```

STEPS TO IMPORT CPP CODE INTO PYTHON CODE:

- **Select File→New** in Notepad and type the above Python program.
- Save the File as **welcome.py**.
- Click the Run Terminal and open the command window
- Go to the folder of Python using cd command.
- Type the command**: Python c:\pyprg\welcome.py -i c:\pyprg\welcome_cpp**

OUTPUT:

```
----------------------------------------
        WELCOME
----------------------------------------
```

PREPARED BY

J. BASKARAN M.Sc., B.Ed. (C.S)
jbaskaran89@gmail.com

J. ILAKKIA M.Sc., M.Phil., B.Ed. (C.S)
jilakkia@gmail.com

J. BASKARAN M.Sc., B.Ed. (C.S)
jbaskaran89@gmail.com
Puducherry.

J. ILAKKIA M.Sc., M.Phil., B.Ed. (C.S)
jilakkia@gmail.com
Puducherry.

COMPUTER SCIENCE

15. DATA MANIPULATION THROUGH SQL

Section – A

Choose the best answer **(1 Mark)**

1. Which of the following is an organized collection of data?

(A) Database (B) DBMS (C) Information (D) Records

2. SQLite falls under which database system?

(A) Flat file database system **(B) Relational Database system**

(C) Hierarchical database system (D) Object oriented Database system

3. Which of the following is a control structure used to traverse and fetch the records of the database?

(A) Pointer (B) Key **(C) Cursor** (D) Insertion point

4. Any changes made in the values of the record should be saved by the command

(A) Save (B) Save As **(C) Commit** (D) Oblige

5. Which of the following executes the SQL command to perform some action?

(A) Execute() (B) Key() (C) Cursor() (D) run()

6. Which of the following function retrieves the average of a selected column of rows in a table?

(A) Add() (B) SUM() **(C) AVG()** (D) AVERAGE()

7. The function that returns the largest value of the selected column is

(A) MAX() (B) LARGE() (C) HIGH() (D) MAXIMUM()

8. Which of the following is called the master table?

(A) sqlite_master (B) sql_master (C) main_master (D) master_main

9. The most commonly used statement in SQL is

(A) cursor **(B) select** (C) execute (D) commit

10. Which of the following clause avoide the duplicate?

(A) Distinct (B) Remove (C) Where (D) GroupBy

Section-B

Answer the following questions **(2 Marks)**

1. Mention the users who uses the Database.

- Users of database can be human users, other programs or applications

2. Which method is used to connect a database? Give an example.

- Create a connection using **connect () method** and pass the name of the database File.
- **Example:**

```
import sqlite3
# connecting to the database
connection = sqlite3.connect ("Academy.db")
# cursor
cursor = connection.cursor()
```

3. What is the advantage of declaring a column as "INTEGER PRIMARY KEY"

- If a column of a table is declared to be an **INTEGER PRIMARY KEY**, then whenever a NULL will be used as an input for this column, the **NULL will be automatically converted into an integer** which will one larger than the highest value so far used in that column.
- If the table is empty, the value 1 will be used.

4. Write the command to populate record in a table. Give an example.

- To populate (add record) the table "INSERT" command is passed to SQLite. "execute" method executes the SQL command to perform some action.
- **Example:**

```
sql_command = """INSERT INTO Student (Rollno, Sname, Grade, gender, Average, birth_date) VALUES (NULL, "Akshay", "B", "M","87.8", "2001-12-12");""" cursor.execute(sql_command)
```

5. Which method is used to fetch all rows from the database table?

- The **fetchall()** method is used to fetch all rows from the database table.
- **Example:** result = cursor.fetchall()

Section-C

Answer the following questions **(3 Marks)**

1. What is SQLite?What is it advantage?

- SQLite is a simple relational database system, which saves its data in regular data files or even in the internal memory of the computer.

ADVANTAGES:

- SQLite is fast, rigorously tested, and flexible, making it easier to work.
- Python has a native library for SQLite.

2. Mention the difference between fetchone() and fetchmany()

fetchone()	fetchmany()
• The **fetchone()** method returns the next row of a query result set or None in case there is no row left	• The **fetchmany()** method returns the next number of rows (n) of the result set.
• Using while loop and fetchone() method we can display all the records from a table.	• Displaying specified number of records is done by using **fetchmany().**

3. What is the use of Where Clause. Give a python statement Using the where clause.

- The WHERE clause is used to extract only those records that fulfill a specified condition.

EXAMPLE: To display the different grades scored by male students from "student table"

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT DISTINCT (Grade) FROM student where gender='M'")
result = cursor.fetchall()
print(*result,sep="\n")
```

OUTPUT:

```
('B',)
('A',)
('C',)
('D',)
```

4. Read the following details.Based on that write a python script to display department wise records.

database name :- organization.db
Table name :- Employee
Columns in the table :- Eno, EmpName, Esal, Dept

PYTHON SCRIPT:

```
import sqlite3
connection = sqlite3.connect("organization.db")
c=conn.execute("SELECT * FROM Employee GROUP BY Dept")
for row in c:
        print(row)
conn.close()
```

5. Read the following details.Based on that write a python script to display records in desending order of Eno.

database name :- organization.db

Table name :- Employee

Columns in the table :- Eno, EmpName, Esal, Dept

PYTHON SCRIPT:

```
import sqlite3
connection = sqlite3.connect("organization.db")
cursor=connection.cursor()
cursor.execute("SELECT * FROM Employee ORDER BY Eno DESC")
result=cursor.fetchall()
print(result)
```

Section - D

Answer the following questions: **(5 Marks)**

1. Write in brief about SQLite and the steps used to use it.

- SQLite is a simple relational database system, which saves its data in regular data files or even in the internal memory of the computer.
- It is designed to be embedded in applications, instead of using a separate database server program such as MySQLor Oracle.

ADVANTAGES:

- SQLite is fast, rigorously tested, and fl exible, making it easier to work.
- Python has a native library for SQLite.

Steps To Use SQLite:

Step 1: import sqlite3

Step 2: Create a connection using connect () method and pass the name of the database File

- Connecting to a database in step2 means passing the name of the database to be accessed.
- If the database already exists the connection will open the same.
- Otherwise, Python will open a new database file with the specified name.

Step 3: Set the cursor object cursor = connection. cursor ()

- Cursor is a control structure used to traverse and fetch the records of the database.
- Cursor has a major role in working with Python.
- All the commands will be executed using cursor object only.
- To create a table in the database, create an object and write the SQL command in it.

Example:- sql_comm = "SQL statement"

- For executing the command use the cursor method and pass the required sql command as a parameter.
- Many number of commands can be stored in the sql_comm and can be executed one after other.
- Any changes made in the values of the record should be saved by the commend **"Commit"** before closing the "Table connection".

2. Write the Python script to display all the records of the following table using fetchmany()

Icode	ItemName	Rate
1003	Scanner	10500
1004	Speaker	3000
1005	Printer	8000
1008	Monitor	15000
1010	Mouse	700

PYTHON SCRIPT:

```
import sqlite3
connection = sqlite3.connect("Materials.db")
cursor=connection.cursor()
cursor.execute("SELECT * FROM Materials")
print("Displaying All The Records")
result=cursor.fetchmany(5)
print(result, Sep= "\n")
```

OUTPUT:

Displaying All The Records

(1003, 'Scanner', 10500)

(1004, 'Speaker', 3000)

(1005, 'Printer', 8000)

(1008, 'Monitor', 15000)

(1010, 'Mouse', 700)

3. What is the use of HAVING clause. Give an example python script

- Having clause is used to filter data based on the group functions.
- This is similar to WHERE condition but can be used only with group functions.
- Group functions cannot be used in WHERE Clause but can be used in HAVING clause.
- **Example:**

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT GENDER,COUNT(GENDER) FROM Student GROUP BY GENDER HAVING COUNT(GENDER)>3")
result = cursor.fetchall()
co = [i[0] for i in cursor.description]
print(co)
print(result)
```

OUTPUT:

```
['gender', 'COUNT(GENDER)']
[('M', 5)]
```

4. Write a Python script to create a table called ITEM with following specification.
Add one record to the table.
Name of the database :- ABC
Name of the table :- Item
Column name and specification :-

Icode :-	**integer and act as primary key**
Item Name :-	**Item Name :-**
Rate :-	**Integer**
Record to be added :-	**1008, Monitor,15000**

PYTHON SCRIPT:

```
import sqlite3
connection = sqlite3.connect("ABC.db")
cursor=connection.cursor()
sql_command – """ CREATE TABLE Item(
Icode INTEGER PRIMARY KEY,
```

ItemName VARCHAR(25),

Rate INTEGER) ; """"

cursor.execute(sql_command)

sql_command = """" INSERT INTO Item(Icode, ItemName, Rate) VALUES (1008, 'Monitor', 15000); """"

cursor.execute(sql_command)

connection.commit()

connection.close()

print("TABLE CREATED")

OUTPUT:

TABLE CREATED

5. Consider the following table Supplier and item .Write a python script for (i) to (ii)

SUPPLIER				
Suppno	**Name**	**City**	**Icode**	**SuppQty**
S001	**Prasad**	**Delhi**	**1008**	**100**
S002	**Anu**	**Bangalore**	**1010**	**200**
S003	**Shahid**	**Bangalore**	**1008**	**175**
S004	**Akila**	**Hydrabad**	**1005**	**195**
S005	**Girish**	**Hydrabad**	**1003**	**25**
S006	**Shylaja**	**Chennai**	**1008**	**180**
S007	**Lavanya**	**Mumbai**	**1005**	**325**

PYTHON SCRIPT:

i) Display Name, City and Itemname of suppliers who do not reside in Delhi.

```
import sqlite3
connection = sqlite3.connect("ABC.db")
cursor.execute("SELECT Supplier.Name, Supplier.City,Item.ItemName FROM Supplier,Item
                WHERE Supplier.Icode = Item.Icode AND Supplier.City NOT In Delhi ")
s = [i[0] for I in cursor.description]
    print(s)
result = cursor.fetchall()
for r in result:
    print r
```

OUTPUT:

['Name',	**'City',**	**'ItemName']**
['Anu',	**'Bangalore',**	**'Scanner']**
['Shahid',	**'Bangalore',**	**'Speaker']**
['Akila',	**'Hydrabad',**	**'Printer']**
['Girish',	**'Hydrabad',**	**'Monitor']**
['Shylaja',	**'Chennai',**	**'Mouse']**
['Lavanya',	**'Mumbai',**	**'CPU']**

ii) Increment the SuppQty of Akila by 40

```
import sqlite3
connection = sqlite3.connect("ABC.db")
cursor.execute("UPDATE Supplier ST SuppQty = SuppQty +40 WHERE Name = 'Akila' ")
cursor.commit()
result = cursor.fetchall()
print (result)
connection.close()
```

OUTPUT:

(S004, 'Akila', 'Hydrabad', 1005, 235)

PREPARED BY

J. BASKARAN M.Sc., B.Ed. (C.S)
jbaskaran89@gmail.com

J. ILAKKIA M.Sc., M.Phil., B.Ed. (C.S)
jilakkia@gmail.com