

VIRUDHUNAGAR DISTRICT  
COMMON HALF YEARLY EXAMINATION - DECEMBER 2024  
STANDARY 12  
COMPUTER SCIENCE

PART - I

I. Answer all and Choose the best answer:

1. a. subroutines
2. b. selectors
3. c. Access control
4. d. Half-interval search
5. d.assignment
6. b.Relational or logical expression
7. d. def
8. b. Multi line strings
9. : (colon)
10. \_\_num
11. c. Hierarchical
12. b. ORDER BY
13. a. Flat file
14. d. HTML
15. c. Commit

PART - II

II. Answer ANY 6 of the following and question no. 23 is compulsory: 6 X 2 = 12

16. What is space time trade off?

A space-time or time-memory trade-off is a way of solving in less time by using more storage space or by solving a given algorithm in very little space by spending more time.

### 17. What are keywords? Give Example.

#### Keywords:

- ❖ Keywords are special words used by Python interpreter to recognize the structure of program.
- ❖ These words have specific meaning for interpreter, they cannot be used for any other purpose.

Example : class, for, if, switch, global,

### 18. Write a note on 'continue' statement in Python.

- ❖ The Continue statement is used to skip the remaining part of a loop and Control of the program flows to the statement immediately after the body of the loop.

### 19. Write about replace() function in python.

- ❖ The replace() method in Python is used to replace a specified substring with another substring within a string.
- ❖ It returns a new string with the replacements applied, while the original string remains unchanged.  
replace("char1", "char2")

### 20. Write a python program using class to accept three sides of a triangle and print its area

Class area:

```
def area (self, b, h):
print ("Area of Triangle",
(b* h)/2)
A = area ()
b=int (input ("enter base'))
h=int (input ("enter height'))
A. area (b, h)
```

### 21. List and explain the component of a database.

The Database Management System can be divided into five major components as follows:

- 1.Hardware
- 2.Software
- 3.Data
- 4.Procedures/Methods
- 5.Database Access Languages

## 22. What is the usage of IN keyword in SQL?

### IN Keyword

- ❖ The **IN** keyword is used to specify a list of values which must be matched with the record values.
- ❖ In other words it is used to compare a column with more than one value.
- ❖ It is similar to an **OR** condition.

For example :

```
SELECT Admno, Name, Place FROM Student WHERE Place IN ('Chennai', 'Delhi');
```

## 23. Explain fetchone() method with an example program.

### fetchone() method:

- ❖ The fetchone() method returns the next row of a query result set or None in case there is no row left.

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT * FROM student")
print("\nfetch one:")
res = cursor.fetchone()
print(res)
```

### OUTPUT

```
fetch one:
(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12')
```

## 24. List some commonly used interfaces for wrapping.

- ❖ 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)

## PART - III

III. Answer ANY 6 of the following and question no. 30 is compulsory:  $6 \times 3 = 18$

25. Mention the characteristics of interface.

**Characteristics of Interface:**

- ❖ The class template specifies the interfaces to enable an object to be created and operated properly.
- ❖ An object's attributes and behaviour is controlled by sending functions to the object.

26. Write note on Asymptotic notation.

**Asymptotic notation:**

- ❖ **Asymptotic Notations** are languages that use meaningful statements about time and space complexity.
- ❖ The following three asymptotic notations are mostly used to represent time complexity of algorithms:

**Big O**

- ❖ Big O is often used to describe the worst-case of an algorithm.

**Big  $\Omega$**

- ❖ Big Omega is the reverse Big O.
- ❖ **Example:** If **Big O** is used to describe the upper bound (worst - case) then, **Big  $\Omega$**  is used to describe the lower bound (best-case).

**Big  $\Theta$**

- ❖ When an algorithm has a complexity with **lower bound = upper bound**, that algorithm has a complexity  $O(n \log n)$  and  $\Omega(n \log n)$ , it's actually has the complexity  $\Theta(n \log n)$ .
- ❖ Time complexity is  **$n \log n$**  in both best-case and worst-case.

27. Write short notes on Arithmetic operators with examples.

**Arithmetic operator:**

- ❖ An arithmetic operator is a mathematical operator used for simple arithmetic.
- ❖ It takes two operands and performs a calculation on them.

Operator - Operation	Examples	Result
Assume a=100 and b=10. Evaluate the following expressions		
+ (Addition)	>>> a + b	110
- (Subtraction)	>>> a - b	90
* (Multiplication)	>>> a * b	1000
/ (Division)	>>> a / b	10.0
% (Modulus)	>>> a % 30	10
** (Exponent)	>>> a ** 2	10000
// (Floor Division)	>>> a // 30 (Integer Division)	3

### 28. Write note on if..else structure.

- ❖ The if .. else statement provides control to check the true block as well as the false block.
- ❖ if..else statement thus provides two possibilities and the condition determines which BLOCK is to be executed.

#### Syntax:

if <condition>:

statements-block 1

else:

statements-block 2

### 29. How recursive function works?

1. Recursive function is called by some external code.
2. If the base condition is met then the program gives meaningful output and exits.
3. Otherwise, function does some required processing and then calls itself to continue recursion.

### 30. What are the difference between List and Tuple?

List	Tuple
The elements of a list are changeable (mutable)	the elements of a tuple are unchangeable (immutable)
The elements of a list are enclosed within square brackets	the elements of a tuple are enclosed by parenthesis

### 31. Explain Cartesian product with a suitable example.

- ❖ Cross product is a way of combining two relations.
- ❖ The resulting relation contains, both relations being combined.
- ❖ This type of operation is helpful to merge columns from two relations.

**Example:**  $A \times B$  means A times B, where the relation A and B have different attributes.

### 32. Write any three DDL commands.

<b>Create</b>	To create tables in the database.
<b>Alter</b>	Alters the structure of the database.
<b>Drop</b>	Delete tables from database.
<b>Truncate</b>	Remove all records from a table, also release the space occupied by those records.

### 33. What is MinGw? What is its use?

#### MinGW:

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++.

### PART - IV

#### 34. a. 1. Explain with example Pure and impure functions.

Pure functions	Impure functions
Pure functions will give exact result when the same arguments are passed.	Impure functions never assure you that the function will behave the same every time it's called.
Pure function does not cause any side effects to its output.	Impure function causes side effects to its output.
The return value of the pure functions solely depends on its arguments passed.	The return value of the impure functions does not solely depend on its arguments passed.
They do not modify the arguments which are passed to them	They may modify the arguments which are passed.
If we call pure functions with same set of arguments, we will always get the same return values.	If we call impure functions with same set of arguments, we might get the different return values.
<b>Example: sqrt()</b> let square x return: $x * x$	<b>Example: random()</b> let Random number let a := random() if a > 10 then return: a else return: 10

OR

Explain the types of scopes for variable or LEGB rule with example.

#### SCOPE:

- ❖ Scope refers to the visibility of variables, parameters and functions in one part of a program to another part of the same program.

**TYPES OF VARIABLE SCOPE:**

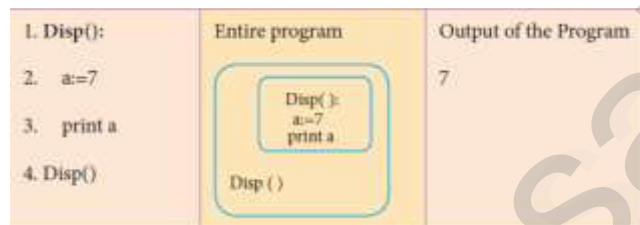
- ❖ Local Scope
- ❖ Enclosed Scope
- ❖ Global Scope
- ❖ Built-in Scope

**LEGB RULE:**

- ❖ The **LEGB** rule is used to decide the order in which the scopes are to be searched for scope resolution.
- ❖ The scopes are listed below in terms of hierarchy (highest to lowest).

**i) LOCAL SCOPE:**

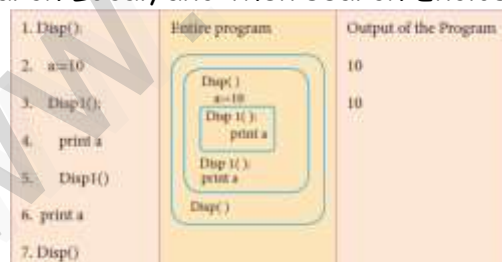
- ❖ Local scope refers to variables defined in current function.
- ❖ A function will always look up for a variable name in its local scope.
- ❖ Only if it does not find it there, the outer scopes are checked.

**Example:**

- ❖ On execution of the above code the variable **a** displays the value 7, because it is defined and available in the local scope.

**ii) ENCLOSED SCOPE:**

- ❖ A variable which is declared inside a function which contains another function definition with in it, the inner function can also access the variable of the outer function. This scope is called enclosed scope.
- ❖ When a compiler or interpreter searches for a variable in a program, it first search Local, and then search Enclosing scopes.



- ❖ In the above example Disp1() is defined within Disp(). The variable „a“ defined in Disp() can be even used by Disp1() because it is also a member of Disp().

**iii) GLOBAL SCOPE:**

- ❖ A variable which is declared outside of all the functions in a program is known as global variable.

- ❖ Global variable can be accessed inside or outside of all the functions in a program.

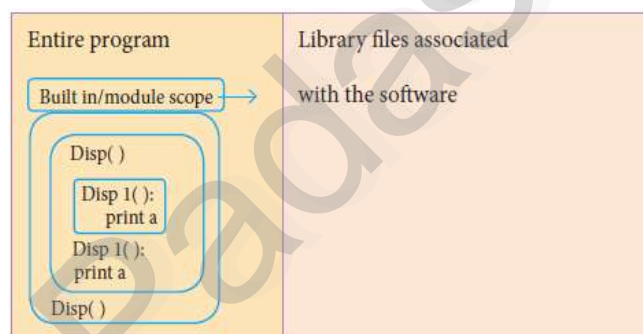
### Example:

	Entire program	Output of the Program
1. a:=10		7 10
2. Disp():		
3. a:=7		
4. print a		
5. Disp()		
6. print a		

- ❖ On execution of the above code the variable **a** which is defined inside the function displays the value 7 for the function call Disp() and then it displays 10, because **a** is defined in global scope.

### iv) BUILT-IN-SCOPE:

- ❖ The built-in scope has all the names that are pre-loaded into the program scope when we start the compiler or interpreter.
- ❖ Any variable or module which is defined in the library functions of a programming language has Built-in or module scope.



### 35. Discuss about linear search algorithm.

#### LINEAR SEARCH:

- ❖ Linear search also called sequential search is a sequential method for finding a particular value in a list.
- ❖ This method checks the search element with each element in sequence until the desired element is found or the list is exhausted.
- ❖ In this searching algorithm, list need not be ordered.

#### Pseudo code:

1. Traverse the array using for loop
2. In every iteration, compare the target search key value with the current value of the list.

- ❖ If the values match, display the current index and value of the array



- ❖ If the values do not match, move on to the next array element. If no match is found, display the search element not found.

3. If no match is found, display the search element not found.

**Example:**

- ❖ To search the number 25 in the array given below, linear search will go step by step in a sequential order starting from the first element in the given array.
- ❖ if the search element is found that index is returned otherwise the search is continued till the last index of the array.
- ❖ In this example number 25 is found at index number 3.

index	0	1	2	3	4
values	10	12	20	25	30

**Snippet:**

Input: values[] = {10,12,20,25,30}

Target=25

**Output:**

3

OR

**Write a detail note on 'for' loop.**

**'for' loop:**

- ❖ **for** loop is the most comfortable loop.
- ❖ It is also an entry check loop.
- ❖ The condition is checked in the beginning and the body of the loop(statements-block 1) is executed if it is only True otherwise the loop is not executed.

**Syntax:**

for counter\_variable in sequence:

statements-block 1

[else: # optional block

statements-block 2]

- ❖ The *counter\_variable* is the control variable.
- ❖ The *sequence* refers to the initial, final and increment value.
- ❖ **for** loop uses the *range()* function in the sequence to specify the initial, final and increment values.
- ❖ **range()** generates a list of values starting from **start** till **stop-1**.

**The syntax of range() is as follows:**

range (start,stop,[step])

Where,

**start** - refers to the initial value

**stop** - refers to the final value

**step** - refers to increment value, this is optional part.

**Example:**

```
for i in range(2,10,2):
    print (i,end=' ')
else:
    print ("\nEnd of the loop")
```

**Output:**

```
2 4 6 8
End of the loop
```

**36. Explain about string operators in python with suitable example.**

**STRING OPERATORS:**

❖ Python provides the following string operators to manipulate string.

**(i) Concatenation (+)**

❖ Joining of two or more strings using plus (+) operator is called as **Concatenation**.

**Example**

```
>>> "welcome" + "Python"
```

**Output:**

```
'welcomePython'
```

**(ii) Append (+ =)**

❖ Adding more strings at the end of an existing string using operator += is known as **append**.

**Example:**

```
>>> str1="Welcome to "
>>> str1+="Learn Python"
>>> print (str1)
```

**Output:**

```
Welcome to Learn Python
```

```
count(str, beg, end
```

**(iii) Repeating (\*)**

❖ The multiplication operator (\*) is used to display a string in multiple number of times.

**Example:**

```
>>> str1="Welcome "
>>> print (str1*4)
```

**Output:**

Welcome Welcome Welcome Welcome

**(iv) String slicing**

- ❖ Slice is a substring of a main string.
- ❖ A substring can be taken from the original string by using [ ] **slicing operator** and index values.
- ❖ Using slice operator, you have to slice one or more substrings from a main string.

**General format of slice operation:**

str[start:end]

- ❖ Where **start** is the beginning index and **end** is the last index value of a character in the string.
- ❖ Python takes the end value less than one from the actual index specified.

**Example: slice a single character from a string**

```
>>> str1="THIRUKKURAL"
```

```
>>> print (str1[0])
```

**Output:**

T

**(v) Stride when slicing string**

- ❖ When the slicing operation, you can specify a third argument as the stride, which refers to the number of characters to move forward after the first character is retrieved from the string.
- ❖ The default value of stride is 1.
- ❖ Python takes the last value as n-1
- ❖ You can also use negative value as stride, to prints data in reverse order.

**Example:**

```
>>> str1 = "Welcome to learn Python"
```

```
>>> print (str1[10:16])
```

```
>>> print(str1[::-2])
```

**Output:**

Learn

nhyre teolW

OR

**Explain the different set operations supported by python with suitable example.**

- ❖ A Set is a mutable and an unordered collection of elements without duplicates.

**Set Operations:**

- ❖ The set operations such as Union, Intersection, difference and Symmetric difference.

**(i) Union:**

- ❖ It includes all elements from two or more sets.
- ❖ The **operator |** is used to union of two sets.
- ❖ The function **union( )** is also used to join two sets in python.

**Example:**

```
set_A={2,4,6,8}
set_B={'A', 'B', 'C', 'D'}
U_set=set_A|set_B
print(U_set)
```

**Output:**

```
{2, 4, 6, 8, 'A', 'D', 'C', 'B'}
```

**(ii) Intersection:**

- ❖ It includes the common elements in two sets.
- ❖ The **operator &** is used to intersect two sets in python.
- ❖ The function **intersection( )** is also used to intersect two sets in python.

**Example:**

```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A & set_B)
```

**Output:**

```
{'A', 'D'}
```

**(iii) Difference:**

- ❖ It includes all elements that are in first set (say set A) but not in the second set (say set B).
- ❖ The minus **(-)** **operator** is used to difference set operation in python.
- ❖ The function **difference( )** is also used to difference operation.

**Example:**

```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A - set_B)
```

**Output:**

```
{2, 4}
```

**(iv) Symmetric difference**

- ❖ It includes all the elements that are in two sets (say sets A and B) but not the one that are common to two sets.
- ❖ The caret **(^)** **operator** is used to symmetric difference set operation in python.
- ❖ The function **symmetric\_difference( )** is also used to do the same operation.

**Example:**

```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
```

```
print(set_A ^ set_B)
```

Output:

```
{2, 4, 'B', 'C'}
```

### 37. Explain the different types of data model.

#### Data Model

- ❖ A data model describes how the data can be represented and accessed from a software after complete implementation

#### Types of Data Model

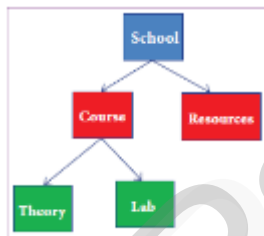
The different types of a Data Model are,

- ❖ Hierarchical Model
- ❖ Relational Model
- ❖ Network Database Model
- ❖ Entity Relationship Model
- ❖ Object Model

#### i). Hierarchical Model:

- ❖ In Hierarchical model, data is represented as a simple tree like structure form.
- ❖ This model represents a one-to-many relationship ie parent-child relationship.
- ❖ One child can have only one parent but one parent can have many children.
- ❖ This model is mainly used in IBM Main Frame computers.

Example:



#### ii). Relational Model

- ❖ The Relational Database model was first proposed by E.F. Codd in 1970 .
- ❖ The basic structure of data in relational model is tables (relations).
- ❖ All the information's related to a particular type is stored in rows of that table.
- ❖ Hence tables are also known as relations in a relational model.
- ❖ A relation key is an attribute which uniquely identifies a particular tuple (row in a relation (table)).

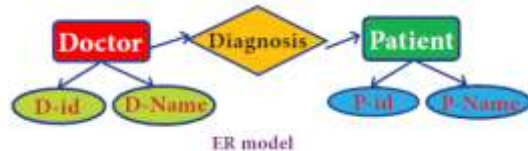
#### iii.) Network Model

- ❖ Network database model is an extended form of hierarchical data model.
- ❖ In a Network model, a child may have many parent nodes.
- ❖ It represents the data in many-to-many relationships.
- ❖ This model is easier and faster to access the data.

#### iv.) Entity Relationship Model. (ER model)

- ❖ In this database model, relationship are created by dividing the object into entity and its characteristics into attributes.
- ❖ It was developed by Chen in 1976.
- ❖ ER model constructed by,
- ❖ **Rectangle** represents the entities.
- ❖ **Ellipse** represents the attributes .
- ❖ **Attributes** describes the characteristics and each entity.
- ❖ **Diamond** represents the relationship in ER diagrams

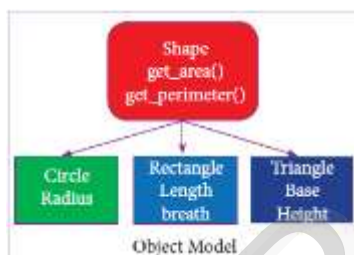
Example: Doctor diagnosis the Patient.



#### v.) Object Model

- ❖ Object model stores the data in the form of objects, attributes and methods, classes and Inheritance.
- ❖ This model handles more complex applications, such as Geographic information System (GIS), scientific experiments, engineering design and manufacturing.

Example:

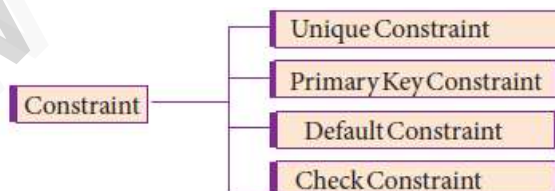


OR

Write different types of constraints and their functions.

- ❖ Constraint is a condition applicable on a field or set of fields.

Type of Constraints:



#### (i) Unique Constraint: Table Constraint

- ❖ This constraint ensures that no two rows have the same value in the specified columns.

For example **UNIQUE** constraint applied on Admno of student table ensures that no two students have the same admission number and the constraint can be used as:

**CREATE TABLE Student**

```
(
Admno integer NOT NULL UNIQUE, → Unique constraint
Name char (20) NOT NULL,
Gender char (1),
);
```

- ❖ The **UNIQUE** constraint can be applied only to fields that have also been declared as **NOT NULL**.
- ❖ When two constraints are applied on a single field, it is known as multiple constraints.
- ❖ In the above Multiple constraints **NOT NULL** and **UNIQUE** are applied on a single field Admno.

**(ii) Primary Key Constraint:**

- ❖ This constraint declares a field as a Primary key which helps to uniquely identify a record.
- ❖ It is similar to unique constraint except that only one field of a table can be set as primary key.
- ❖ The primary key does not allow **NULL** values and therefore a field declared as primary key must have the **NOT NULL** constraint.

**Example:****CREATE TABLE Student**

```
(
Admno integer NOT NULL PRIMARY KEY, → Primary Key constraint
Name char(20)NOT NULL,
Gender char(1),
Age integer,
);
```

**(iii) DEFAULT Constraint:**

- ❖ The **DEFAULT** constraint is used to assign a default value for the field.
- ❖ When no value is given for the specified field having **DEFAULT** constraint, automatically the default value will be assigned to the field.

**Example:****CREATE TABLE Student**

```
(
Admno integer NOT NULL PRIMARY KEY,
Name char(20)NOT NULL,
Gender char(1),
Age integer DEFAULT = "17", → Default Constraint
Place char(10));
```

- ❖ In the above example the "Age" field is assigned a default value of 17, therefore when no value is entered in age by the user, it automatically assigns 17 to Age.

**(iv) Check Constraint:**

- ❖ This constraint helps to set a limit value placed for a field.
- ❖ When we define a check constraint on a single column, it allows only the restricted values on that field.

**Example:**

**CREATE TABLE Student**

```
(
Admno integer NOT NULL PRIMARY KEY
Name char(20)NOT NULL,
Gender char(1),
Age integer (CHECK<=19), → Check Constraint
Place char(10),
);
```

- ❖ In the above example the check constraint is set to Age field where the value of Age must be less than or equal to 19.

**(V) Table Constraint:**

- ❖ When the constraint is applied to a group of fields of the table, it is known as Table constraint.
- ❖ The table constraint is normally given at the end of the table definition.
- ❖ Let us take a new table namely Student1 with the following fields Admno, Firstname, Lastname, Gender, Age, Place:

**Example:**

**CREATE TABLE Student 1**

```
(
Admno integer NOT NULL,
Firstname char(20),
Lastname char(20),
Gender char(1),
Age integer,
Place char(10),
PRIMARY KEY (Firstname, Lastname) → Table constraint
);
```

- ❖ In the above example, the two fields, Firstname and Lastname are defined as Primary key which is a Table constraint.



38. Tabulate different modes in CSV file 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'	Open in text mode. (default)
'b'	Open in binary mode.
'+'	Open a file for updating (reading and writing)

OR

What is the purpose of sys, os, getopt modules 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++“ + <variable_name1> „-<mode>“ + <variable_name2>`

<b>os.system</b>	function system() defined in os module
<b>g++</b>	General compiler to compile C++ program under Windows Operating system.
<b>variable_name1</b>	Name of the C++ file without extension .cpp in string format
<b>mode</b>	To specify input or output mode. Here it is o prefixed with hyphen.

**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])
```

**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.
```

- ❖ If args is displayed using print() command it displays the output as [].

**Example:**

```
>>>print(args)
```

```
[]
```

\*\*\*\*\*

Mrs. GEETHAMARIMUTHU, M.Sc., B.Ed.

P.G. Computer Science

VMG RR SRI SARADA SAKTHI MHSS

VIRUDHUNAGAR