## HIGHER SECONDARY – SECOND YEAR

# XII - COMPUTER SCIENCE REDUCED SYLLABUS MINIMUM STUDY MATERIAL 2021-2022

J.ILAKKIA M.Sc., B.Ed., M.Phil.
J.BASKARAN M.Sc., B.Ed.
CS Knowledge Opener
EDUCATION
KNOWLEDGE OPENING &
KNOWLEDGE TRANSFORMATION

| | |
|---|---|
| NAME | |
| STD & GROUP | |

PREPARED BY

J. ILAKKIA M.Sc., B.Ed., M.Phil.
**Computer Instructor Grade-I**
Govt. Hr. Sec. School
V.Pagandai, Villupuram 605 501.

# TABLE OF CONTENTS
## COMPUTER SCIENCE – II YEAR

# 1. FUNCTIONS

## Section – A

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

1. The small sections of code that are used to perform a particular task is called

**(A) Subroutines** (B) Files (C) Pseudo code (D) Modules

2. Which of the following is a unit of code that is often defined within a greater code structure?

(A) Subroutines **(B) Function** (C) Files (D) Modules

3. Which of the following is a distinct syntactic block?

(A) Subroutines (B) Function **(C) Definition** (D) Modules

4. The variables in a function definition are called as

(A) Subroutines (B) Function (C) Definition **(D) Parameters**

5. The values which are passed to a function definition are called

**(A) Arguments** (B) Subroutines (C) Function (D) Definition

6. Which of the following are mandatory to write the type annotations in the function definition?

(A) Curly braces **(B) Parentheses** (C) Square brackets (D) indentations

## Section-B

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

**1. What is a subroutine?**

- Subroutines are small sections of code that are used to perform a particular task that can be used repeatedly.

**2. Define Function with respect to Programming language.**

- A function is a unit of code that is often defined within a greater code structure.
- A function works on many kinds of inputs and produces a concrete output

**3. Write the inference you get from X:=(78).**

- **X:=(78)** is a function definition.
- Definitions bind values to names.
- Hence, the value **78** bound to the name **'X'**.

**5. Which of the following is a normal function definition and which is recursive function definition?**

**i)** let rec sum x y:

return x + y

**Ans: Recursive Function**

**ii)** let disp :

print 'welcome'

**Ans: Normal Function**

**iii)** let rec sum num:

if (num!=0) then return num + sum (num-1)

```
else
return num
```

**Ans: Recursive Function**

## Section - D

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

**1. What are called Parameters and write a note on**

(i) Parameter without Type (ii) Parameter with Type

**Answer:**

- **Parameters** are the variables in a function definition
- **Arguments** are the values which are passed to a function definition.
- Two types of parameter passing are,

1. Parameter Without Type
2. Parameter With Type

### 1. Parameter Without Type:

- Lets see an example of a function definition of Parameter Without Type:

```
(requires: b>=0 )
(returns: a to the power of b)
let rec pow a b:=
        if b=0 then 1
        else a * pow a (b-1)
```

- In the above function definition **variable ‘ b’** is the **parameter** and the **value** passed **to** the variable ***‘b’*** is the **argument**.

### 2. Parameter With Type:

- Now let us write the same function definition with types,

```
(requires: b> 0 )
(returns: a to the power of b )
let rec pow (a: int) (b: int) : int :=
        if b=0 then 1
        else a * pow b (a-1)
```

- In this example we have explicitly annotating the types of argument and return type as ***‘int’***.

# 2. DATA ABSTRACTION

## Section – A

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

1. Which of the following functions that build the abstract data type ?

**(A) Constructors** (B) Destructors (C) recursive (D)Nested

2. Which of the following functions that retrieve information from the data type?

(A) Constructors **(B) Selectors** (C) recursive (D)Nested

6. The data type whose representation is unknown are called

(A) Built in datatype (B) Derived datatype

(C) Concrete datatype **(D) Abstract datatype**

## Section-B

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

**1. What is abstract data type?**

- Abstract Data type (ADT) is a type or class for objects whose behavior is defined by a set of value and a set of operations.

**2. Differentiate constructors and selectors.**

| CONSTRUCTORS | SELECTORS |
|---|---|
| • Constructors are functions that build the abstract data type. | • Selectors are functions that retrieve information from the data type. |
| • Constructors create an object, bundling together different pieces of information | • Selectors extract individual pieces of information from the object. |

## Section-C

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

**3. Identify Which of the following are constructors and selectors?**

| | | |
|---|---|---|
| (a) N1=number() | -- | **Constructor** |
| (b) accetnum(n1) | -- | **Selector** |
| (c) displaynum(n1) | -- | **Selector** |
| (d) eval(a/b) | -- | **Selector** |
| (e) x,y= makeslope (m), makeslope(n) | -- | **Constructor** |
| (f) display() | -- | **Selector** |

## Section - D

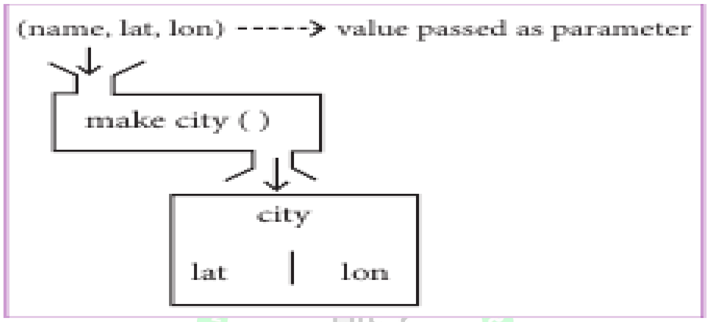
**Answer the following questions: (5 Mark)**

**1. How will you facilitate data abstraction? Explain it with suitable example.**

- Data abstraction is supported by defining an abstract data type (ADT), which is a collection of constructors and selectors.
- To facilitate data abstraction, you will need to create two types of functions:
  - **Constructors**
  - **Selectors**

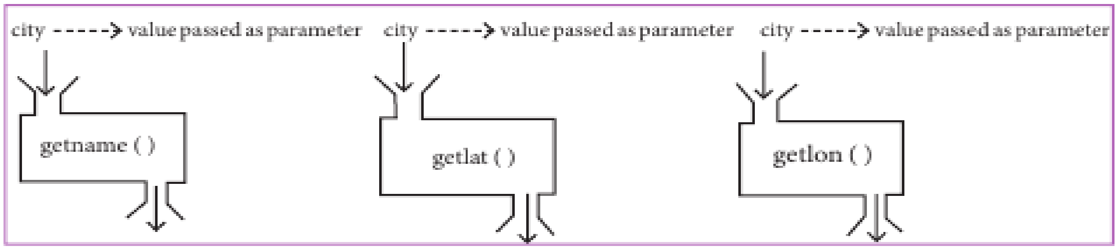
**a) Constructor:**

- Constructors are functions that build the abstract data type.
- Constructors create an object, bundling together different pieces of information.
- For example, say you have an abstract data type called city.



**b) Selectors:**

- Selectors are functions that retrieve information from the data type.
- Selectors extract individual pieces of information from the object.
- To extract the information of a city object, you would use functions like
  - **getname(city)**
  - **getlat(city)**
  - **getlon(city)**


city -----> value passed as parameter
getname ( )
city -----> value passed as parameter
getlat ( )
city -----> value passed as parameter
getlon ( )

## 3. SCOPING

## Section – A

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

1. Which of the following refers to the visibility of variables in one part of a program to another part of the same program.

**(A) Scope** (B) Memory (C) Address (D) Accessibility

2. The process of binding a variable name with an object is called

(A) Scope **(B) Mapping** (C) late binding (D) early binding

3. Which of the following is used in programming languages to map the variable and object?

(A) :: **(B) :=** (C) = (D) ==

4. Containers for mapping names of variables to objects is called

(A) Scope (B) Mapping (C) Binding **(D) Namespaces**

5. Which scope refers to variables defined in current function?

**(A) Local Scope** (B) Global scope (C) Module scope (D) Function Scope

## Section-B

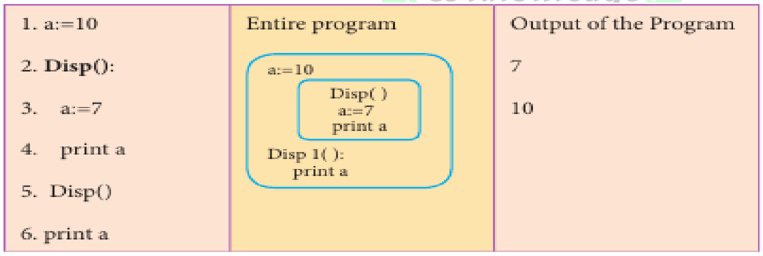
**Answer the following questions (2 Mark)**

**1. What is a scope?**

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

**2. Why scope should be used for variable. State the reason.**

- The scope should be used for variables because; it limits a variable's scope to a single definition.
- That is the variables are visible only to that part of the code.
- **Example:**



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

**3. What is Mapping?**

- The process of binding a variable name with an object is called mapping.
- **:= (colon equal to sign)** is used in programming languages to map the variable and object.

**4. What do you mean by Namespaces?**

- Namespaces are containers for mapping names of variables to objects (name : = object).
- **Example:** **a:=5**
- Here the variable 'a' is mapped to the value '5'.

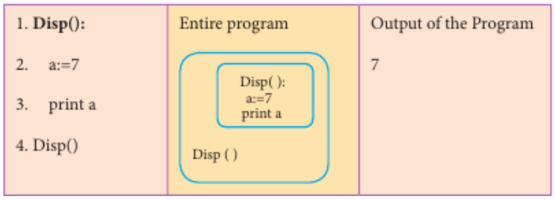
## Section-C

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

**1. Define Local scope with an example.**

- 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:**

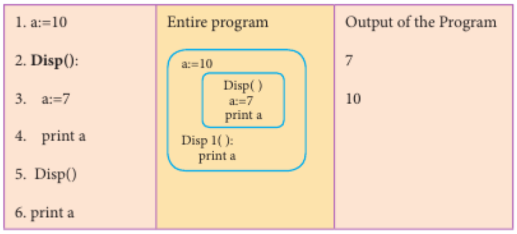


```
Disp():
    a:=7
    print a
Disp()
```

.

**2. Define Global scope with an example.**

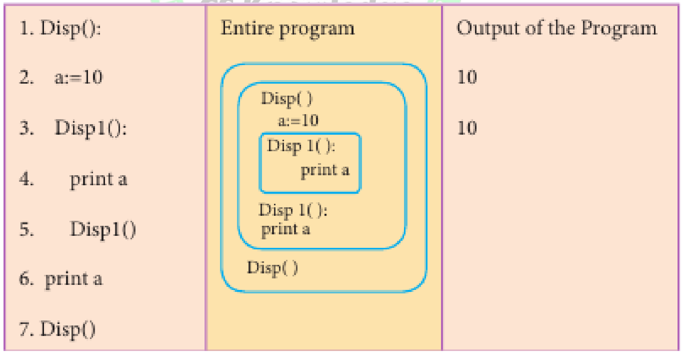
- A variable which is declared outside of all the functions in a program is known as global variable.
- Global variable can be accessed inside or outside of all the functions in a program.
- **Example:**



```
a:=10
Disp():
    a:=7
    print a
Disp()
print a
```

**3. Define Enclosed scope with an example.**

- 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.



```
Disp():
    a:=10
    Disp1():
        print a
        Disp1()
print a
Disp()
```

**5. Identify the scope of the variables in the following pseudo code and write its output.**

```
color:= Red
mycolor():
b:=Blue
myfavcolor():
g:=Green
print color, b, g
myfavcolor()
print color, b
mycolor()
print color
```

**OUTPUT:**

```
Red Blue Green
Red Blue
Red
```

**Scope of Variables:**

| Variables | Scope |
|---|---|
| Color:=Red | Global |
| b:=Blue | Enclosed |
| G:=Green | Local |

## Section - D

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

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

**SCOPE:**

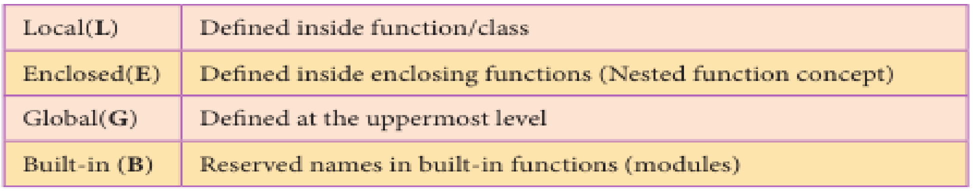
- Scope refers to the visibility of variables, parameters and functions in one part of a program to another part of the same program.

**TYPES OF VARIABLE SCOPE:**

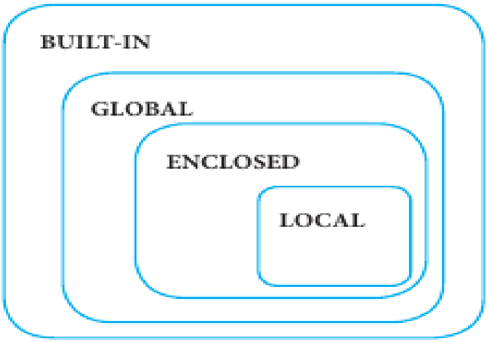
- Local Scope
- Enclosed Scope
- Global Scope
- Built-in Scope

**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).



| | |
|---|---|
| Local(L) | Defined inside function/class |
| Enclosed(E) | Defined inside enclosing functions (Nested function concept) |
| Global(G) | Defined at the uppermost level |
| Built-in (B) | Reserved names in built-in functions (modules) |



**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.

**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.

**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.

**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.

# 4. ALGORITHMIC STRATEGIES

## Section – A

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

1. The word comes from the name of a Persian mathematician Abu Ja'far Mohammed ibn-i Musa al Khowarizmi is called?

(A) Flowchart (B) Flow **(C) Algorithm** (D) Syntax

2. From the following sorting algorithms which algorithm needs the minimum number of swaps?

(A) Bubble sort (B) Quick sort (C) Merge sort **(D) Selection sort**

6. Which of the following is not a stable sorting algorithm?

(A) Insertion sort **(B) Selection sort** (C) Bubble sort (D) Merge sort

## Section-B

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

**1. What is an Algorithm?**

- An algorithm is a finite set of instructions to accomplish a particular task.
- It is a step-by-step procedure for solving a given problem.

**2. Define Pseudo code.**

- **Pseudo code** is a methodology that allows the programmer to represent the implementation of an algorithm.
- It has no syntax like programming languages and thus can't be compiled or interpreted by the computer.

**3. Who is an Algorist?**

- An Algorist is a person skilled in the design of algorithms
- An algorithmic artist

**4. What is Sorting?**

- Sorting is a process of arranging group of items in an ascending or descending order.

**5. What is searching? Write its types.**

- A Search algorithm is the step-by-step procedure used to locate specific data among a collection of data.
- **Example:** Linear Search, Binary Search

## Section-C

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

**1. List the characteristics of an algorithm.**

- Input
- Output
- Finiteness
- Definiteness
- Effectiveness
- Correctness
- Simplicity
- Unambiguous
- Feasibility
- Portable
- Independent

## Section - D

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

**1. Explain the characteristics of an algorithm.**

| Characteristics | Meaning |
|---|---|
| Input | Zero or more quantities to be supplied. |
| Output | At least one quantity is produced. |
| Finiteness | Algorithms must terminate after finite number of steps. |
| Definiteness | All operations should be well defined. |
| Effectiveness | Every instruction must be carried out effectively. |
| Correctness | The algorithms should be error free. |
| Simplicity | Easy to implement. |
| Unambiguous | Algorithm should be clear and unambiguous. |
| Feasibility | Should be feasible with the available resources. |

**2. 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.
3. If no match is found, display the search element not found.
```

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

**Snippet:**

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

Target=25

**Output:**

3

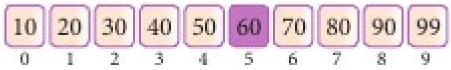
**3. What is Binary search? Discuss with example.**

**BINARY SEARCH:**

- Binary search also called half-interval search algorithm.
- It finds the position of a search element within a sorted array.
- The binary search algorithm can be done as divide-and-conquer search algorithm and executes in logarithmic time.

**Binary Search Working principles with example**:

- Let us assume that the **search element is 60** and we need to search the location or index of search element 60 using binary search.

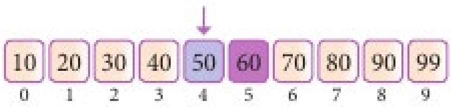


| 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 99 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

- First, we find index of middle element of the array by using this formula :

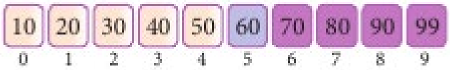
**mid = low + (high - low) / 2**

- Here it is, 0 + (9 - 0 ) / 2 = 4. So, 4 is the mid value of the array.

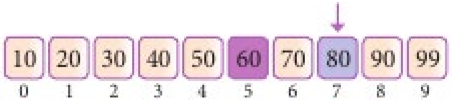


| 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 99 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

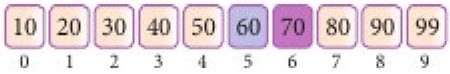
- Compare the value stored at index 4 with target value, which is not match with search element. As the search value 60 > 50.



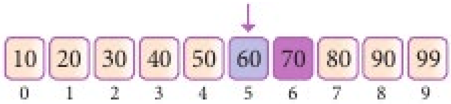
- Now we change our search range **low to mid + 1** and find the new mid value as index 7.
- We compare the value stored at index 7 with our target value.



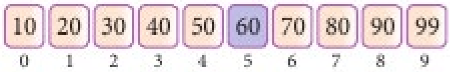
- Element not found because the value in index 7 is greater than search value . ( 80 > 60)
- So, the search element must be in the lower part from the current mid value location



- Now we change our search range **low to mid - 1** and find the new mid value as index 5



- Now we compare the value stored at location 5 with our search element.
- We found that it is a match.



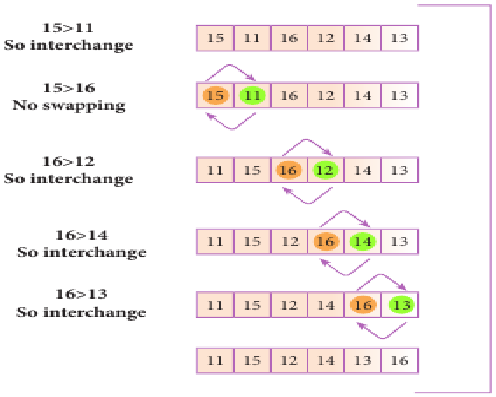
- We can conclude that the search element 60 is found at location or index 5.

**4. Explain the Bubble sort algorithm with example.**

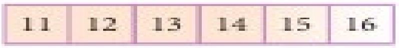
- Bubble sort is a simple sorting algorithm; it starts at the beginning of the list of values stored in an array.
- It compares each pair of adjacent elements and swaps them if they are in the unsorted order.
- This comparison and passed to be continued until no swaps are needed, which shows the values in an array is sorted.
- It is named so because, the smaller elements "bubble" to the top of the list.
- It is too slow and less efficient when compared to other sorting methods.

**Example:**

- Consider an array with values {15, 11, 16, 12, 14, 13}



- At the end of all the iterations we will get the sorted values in an array as given below:



# 5. PYTHON - VARIABLES AND OPERATORS

## Section – A

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

1. Who developed Python ?

A) Ritche **B) Guido Van Rossum** C) Bill Gates D) Sunder Pitchai

2. The Python prompt indicates that Interpreter is ready to accept instruction.

**A) >>>** B) <<< C) # D) <<

3. Which of the following shortcut is used to create new Python Program ?

A) Ctrl + C B) Ctrl + F C) Ctrl + B **D) Ctrl + N**

4. Which of the following character is used to give comments in Python Program ?

**A) #** B) & C) @ D) $

5. This symbol is used to print more than one item on a single line.

A) Semicolon(;) B) Dollor($) **C) comma(,)** D) Colon(:)

6. Which of the following is not a token ?

**A) Interpreter** B) Identifiers C) Keyword D) Operators

7. Which of the following is not a Keyword in Python ?

A) break B) while C) continue **D) operators**

8. Which operator is also called as Comparative operator?

A) Arithmetic **B) Relational** C) Logical D) Assignment

9. Which of the following is not Logical operator?

A) and B) or C) not **D) Assignment**

10. Which operator is also called as Conditional operator?

**A) Ternary** B) Relational C) Logical D) Assignment

## Section-B

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

**1. What are the different modes that can be used to test Python Program ?**

- Interactive mode allows us to write codes in Python command prompt ( >>> ).
- Script mode is used to create and edit python source file with the extension .py

**2. Write short notes on Tokens.**

- Python breaks each logical line into a sequence of elementary lexical components known as Tokens.
- The normal token types are ,

  1) Identifiers,

  2) Keywords,

  3) Operators,

4) Delimiters and

5) Literals.

**3. What are the different operators that can be used in Python ?**

- Operators are special symbols which represent computations, conditional matching in programming.
- Operators are categorized as Arithmetic, Relational, Logical, Assignment and Conditional.

**4. What is a literal? Explain the types of literals ?**

- Literal is a raw data given in a variable or constant.

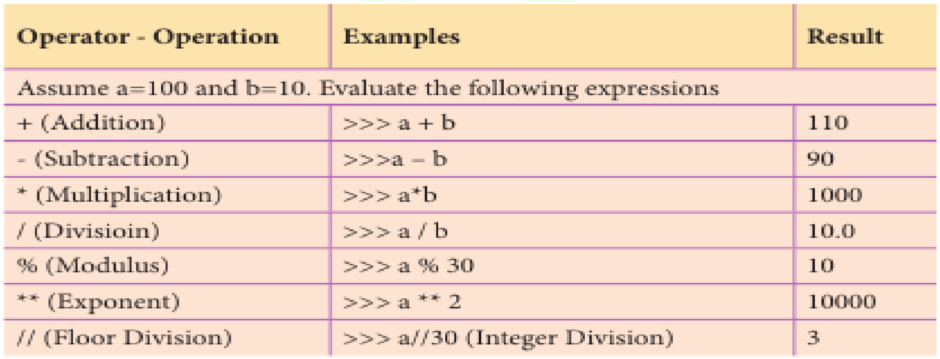
1) **Numeric Literals** consists of digits and are immutable
2) **String literal** is a sequence of characters surrounded by quotes.
3) **Boolean literal** can have any of the two values: True or False.

## Section-C

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

**1. Write short notes on Arithmetic operator with examples.**

- An arithmetic operator is a mathematical operator used for simple arithmetic.
- It takes two operands and performs a calculation on them.
- **Arithmetic Operators used in python:**



| 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 |
| / (Divisioin) | >>> a / b | 10.0 |
| % (Modulus) | >>> a % 30 | 10 |
| ** (Exponent) | >>> a ** 2 | 10000 |
| // (Floor Division) | >>> a//30 (Integer Division) | 3 |

**2. What are the assignment operators that can be used in Python?**

- '=' is a simple **assignment operator** to assign values to variable.
- There are various **compound operators** in Python like **+=, -=, *=, /=, %=, **= and //=.**
- **Example:**

  **a=5**
  **a,b=5,10**
  **a+=2**

**3. Explain Ternary operator with examples.**

- Ternary operator is also known as **conditional operator** that evaluates something based on a condition being true or false.
- It simply allows testing a condition in a single line replacing the multiline if-else making the code compact.

**Syntax:**

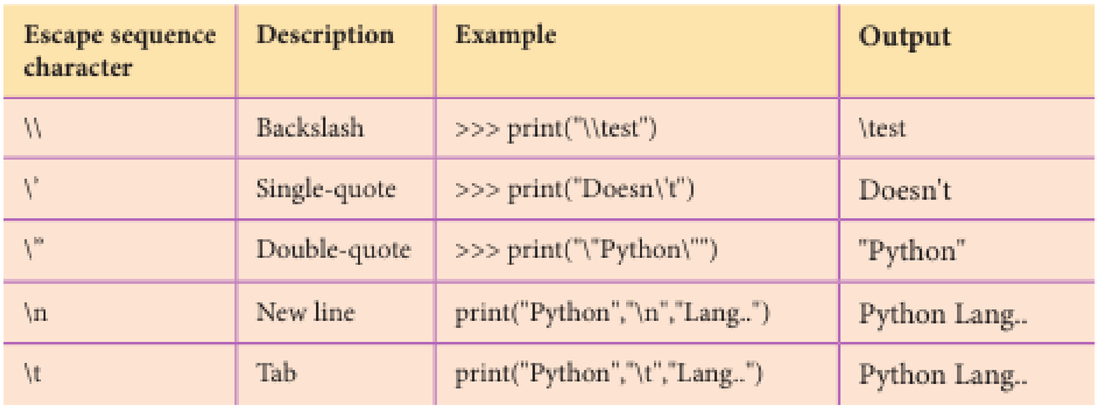
***Variable Name = [on_true] if [Test expression] else [on_false]***

**Example :**

```
min = 50 if 49<50 else 70            # Output: min = 50
```

**4. Write short notes on Escape sequences with examples.**

- In Python strings, the backslash "\" is a special character, also called the **"escape"** character.
- It is used in representing certain whitespace characters.
- Python supports the following escape sequence characters.



| Escape sequence character | Description | Example | Output |
|---|---|---|---|
| \\ | Backslash | >>> print("\\test") | \test |
| \' | Single-quote | >>> print("Doesn\'t") | Doesn't |
| \" | Double-quote | >>> print("\"Python\"") | "Python" |
| \n | New line | print("Python","\n","Lang..") | Python Lang.. |
| \t | Tab | print("Python","\t","Lang..") | Python Lang.. |

**5. What are string literals? Explain.**

- In Python a string literal is a **sequence of characters** surrounded by **quotes**.
- Python supports **single, double and triple quotes** for a string.
- A character literal is a **single character** surrounded by **single or double quotes**.
- The value with **triple-quote ''' '''** is used to give **multi-line** string literal.
- **Example:**

```
strings = "This is Python"
char = "C"
multiline_str = ''' This is a multiline string with more than one line code.'''
print (strings)
print (char)
print (multiline_str)
```

- **Output:**

```
This is Python
C
This is a multiline string with more than one line code.
```

## Section - D

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

**1. Describe in detail the procedure Script mode programming.**

**SCRIPT MODE PROGRAMMING:**

- A script is a text file containing the Python statements.
- Once the Python Scripts is created, they are reusable , it can be executed again and again without retyping.
- The Scripts are editable.

**(i) Creating Scripts in Python**

1. Choose **File → New File** or press **Ctrl + N** in Python shell window.
2. An **untitled** blank script text editor will be displayed on screen.
3. Type the code in Script editor as given below,

**(ii) Saving Python Script**

(1) Choose **File → Save** or Press **Ctrl + S**

(2) Now, **Save As** dialog box appears on the screen.

(3) In the **Save As** dialog box

(4) Finally, click **Save** button to save your Python script.

**(iii) Executing Python Script**

(1) Choose **Run → Run Module** or Press **F5**

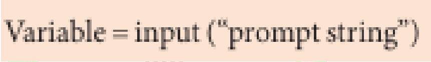
(2) If your code has any error, it will be shown in red color in the IDLE window, and Python describes the type of error occurred.

(3) For all error free code, the output will appear in the IDLE window of Python as shown in **Figure.**

**2. Explain input() and print() functions with examples.**

**1) Input() function**

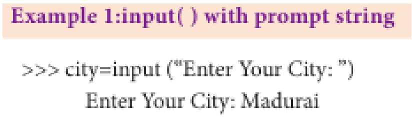
- In Python, **input( )** function is used to accept data as input at run time.
- The syntax for **input()** function is,



```
Variable = input ("prompt string")
```

- **"Prompt string"** in the syntax is a message to the user, to know what input can be given.
- If a prompt string is used, it is displayed on the monitor; the user can provide expected data from the input device.
- The **input( )** takes typed data from the keyboard and stores in the given variable.
- If prompt string is not given in **input( )**, the user will not know what is to be typed as input.

- **Example:**

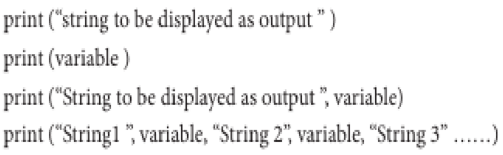


Example 1:input( ) with prompt string

```
>>> city=input ("Enter Your City: ")
    Enter Your City: Madurai
```

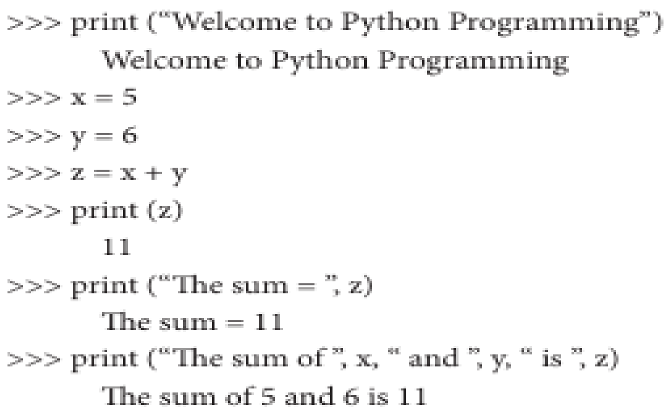
**2) Print() function**

- In Python, the **print()** function is used to display result on the screen.
- **Syntax for print():**



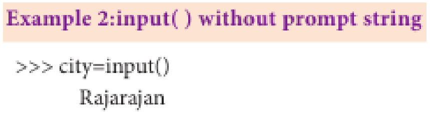
```
print ("string to be displayed as output " )
print (variable )
print ("String to be displayed as output ", variable)
print ("String1 ", variable, "String 2", variable, "String 3" ……)
```

- **Example:**



```
>>> print (“Welcome to Python Programming”)
        Welcome to Python Programming
>>> x = 5
>>> y = 6
>>> z = x + y
>>> print (z)
        11
>>> print (“The sum = ”, z)
        The sum = 11
>>> print (“The sum of ”, x, “ and ”, y, “ is ”, z)
        The sum of 5 and 6 is 11
```

- The **print ( )** evaluates the expression before printing it on the monitor.
- The print () displays an entire statement which is specified within print ( ).
- **Comma ( , )** is used as a separator in **print ( )** to print more than one item.



Example 2:input( ) without prompt string

```
>>> city=input()
     Rajarajan
```

**3. Discuss in detail about Tokens in Python.**

**Tokens**

- Python breaks each logical line into a sequence of elementary lexical components known as **Tokens**.
- The normal token types are,
  1) Identifiers,
  2) Keywords,
  3) Operators,
  4) Delimiters and
  5) Literals.
- Whitespace separation is necessary between tokens, identifiers or keywords.

**1) Identifiers**

- An Identifier is a name used to identify a variable, function, class, module or object.
- An identifier must start with an alphabet (A..Z or a..z) or underscore ( _ ).
- Identifiers may contain digits (0 .. 9)
- Identifiers must not be a **python** keyword.

**2) Keywords**

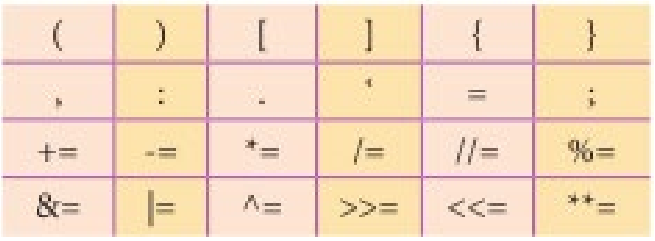
- Keywords are special words used by Python interpreter to recognize the structure of program.
- Keywords have **specific meaning for interpreter**, they cannot be used for any other purpose.

**3) Operators**

- **Operators are special symbols** which represent computations, conditional matching in programming.
- Operators are categorized as Arithmetic, Relational, Logical, Assignment and Conditional.
- Value and variables when used with operator are known as **operands**.

**4) Delimiters**

- Python uses the symbols and symbol combinations as delimiters in expressions, lists, dictionaries and strings.
- Following are the delimiters.



| | | | | | |
|---|---|---|---|---|---|
| ( | ) | [ | ] | { | } |
| , | : | . | ' | = | ; |
| += | -= | *= | /= | //= | %= |
| &= | \|= | ^= | >>= | <<= | **= |

**5) Literals**

- Literal is a raw data given in a variable or constant.
- In Python, there are various types of literals. They are,
  1) **Numeric Literals** consists of digits and are immutable
  2) **String literal** is a sequence of characters surrounded by quotes.
  3) **Boolean literal** can have any of the two values: True or False.

# 6. CONTROL STRUCTURES

## Section – A

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

1. How many important control structures are there in Python?

   **A) 3** B) 4 C) 5 D) 6

2. elif can be considered to be abbreviation of

   A) nested if B) if..else **C) else if** D) if..elif

3. What plays a vital role in Python programming?

   A) Statements B) Control C) Structure **D) Indentation**

4. Which statement is generally used as a placeholder?

   A) continue B) break **C) pass** D) goto

5. The condition in the if statement should be in the form of

   A) Arithmetic or Relational expression B) Arithmetic or Logical expression

   **C) Relational or Logical expression** D) Arithmetic

6. Which is the most comfortable loop?

   A) do..while B) while **C) for** D) if..elif

7. What is the output of the following snippet?

```
i=1
while True:
if i%3 ==0:
break
```

```
    print(i,end='')
  i +=1
```

**A) 1 2**  B) 123  C) 1234  D) 124

8. What is the output of the following snippet?

```
    T=1
    while T:
    print(True)
  break
```

A) False  **B) True**  C) 0  D) no output

9. Which amongst this is not a jump statement ?

**A) for**  B) goto  C) continue  D) break

10. Which punctuation should be used in the blank?

```
    if <condition>_
    statements-block 1
    else:
  statements-block 2
```

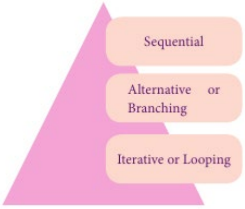
A) ;  **B) :**  C) ::  D) !

## Section-B

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

**1. List the control structures in Python.**

- Three important control structures are,



**2. Write note on break statement.**

**break statement :**

- The **break** statement terminates the loop containing it.
- Control of the program flows to the statement immediately after the body of the loop.

**3. Write is the syntax of if..else statement**

**Syntax:**

```
    if <condition>:
            statements-block 1
    else:
            statements-block 2
```

**4. Define control structure.**

- A program statement that causes a jump of control from one part of the program to another is called control structure or control statement.

**5. Write note on range () in loop**

- range() generates a list of values starting from start till stop-1 in for loop.
- The syntax of range() is as follows:

```
range (start,stop,[step])
```

## Section-C

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

**1. Write a program to display**

**A**

**A B**

**A B C**

**A B C D**

**A B C D E**

**CODE:**

```
for i in range(65, 70):
        for j in range(65, i+1):
                print(chr(j), end= ‘ ‘)
        print(end=’\n’)
        i+=1
```

**OUTPUT**

```
A

A B

A B C

A B C D

A B C D E
```

**2. 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
```

**3. Using if..else..elif statement write a suitable program to display largest of 3 numbers.**

**CODE:**

```
n1= int(input("Enter the first number:"))
n2= int(input("Enter the second number:"))
n3= int(input("Enter the third number:"))
if(n1>=n2)and(n1>=n3):
   biggest=n1;
elif(n2>=n1)and(n2>=n3):
   biggest=n2
else:
   biggest=n3
print("The biggest number between",n1,",",n2,"and",n3,"is",biggest)
```

**OUTPUT**

```
Enter the first number:1
Enter the second number:3
Enter the third number:5
The biggest number between 1 , 3 and 5 is 5
```

**4. Write the syntax of while loop.**

**Syntax:**

```
while <condition>:
    statements block 1
[else:
    statements block2]
```

**5. List the differences between break and continue statements.**

| break | continue |
|---|---|
| The **break** statement terminates the loop containing it. | 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. | Control of the program flows start with next iteration. |
| **Syntax:**<br>break | **Syntax:**<br>continue |

## Section - D

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

**1. Write a detail note on 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:
        statements-block 2]
```

**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.

**2. Write a detail note on if..else..elif statement with suitable example.**

**Nested if..elif...else statement:**

- When we need to construct a chain of **if** statement(s) then **'elif'** clause can be used instead of **'else'.**
- **'elif'** clause combines **if..else-if..else** statements to one **if..elif…else.**
- **'elif'** can be considered to be abbreviation of **'else if'.**
- In an **'if'** statement there is no limit of **'elif'** clause that can be used, but an **'else'** clause if used should be placed at the end.

**Syntax:**

```
    if <condition-1>:
        statements-block 1
    elif <condition-2>:
        statements-block 2
    else:
        statements-block n
```

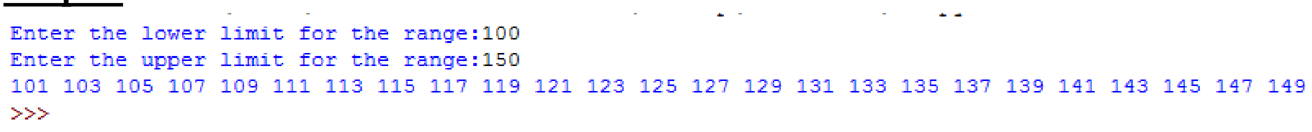
- In the syntax of **if..elif..else** mentioned above, condition-1 is tested if it is true then statements-block1 is executed.

**3. Write a program to display all 3 digit odd numbers.**

**CODE:**

```
lower=int(input("Enter the lower limit for the range:"))
upper=int(input("Enter the upper limit for the range:"))
for i in range(lower,upper+1):
   if(i%2!=0):
      print(i,end=" ")
```

**Output:**



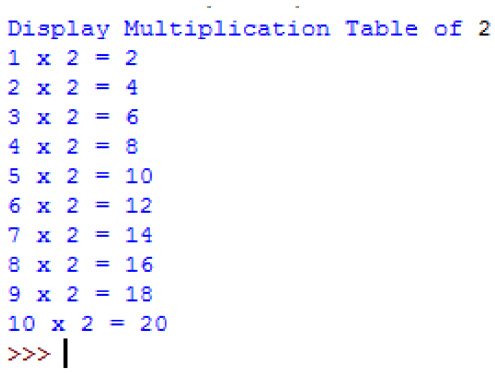
```
Enter the lower limit for the range:100
Enter the upper limit for the range:150
101 103 105 107 109 111 113 115 117 119 121 123 125 127 129 131 133 135 137 139 141 143 145 147 149
>>>
```

**4. Write a program to display multiplication table for a given number.**

**CODE:**

```
num=int(input("Display Multiplication Table of "))
for i in range(1,11):
   print(i, 'x' ,num, '=' , num*i)
```

**Output:**



```
Display Multiplication Table of 2
1 x 2 = 2
2 x 2 = 4
3 x 2 = 6
4 x 2 = 8
5 x 2 = 10
6 x 2 = 12
7 x 2 = 14
8 x 2 = 16
9 x 2 = 18
10 x 2 = 20
>>> |
```

# 7. PYTHON FUNCTIONS

## Section – A

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

1. A named blocks of code that are designed to do one specific job is called as

(a) Loop (b) Branching **(c) Function** (d) Block

2. A Function which calls itself is called as

(a) Built-in **(b) Recursion** (c) Lambda (d) return

3. Which function is called anonymous un-named function

**(a) Lambda** (b) Recursion (c) Function (d) define

4. Which of the following keyword is used to begin the function block?

(a) define (b) for (c) finally **(d) def**

5. Which of the following keyword is used to exit a function block?

(a) define **(b) return** (c) finally (d) def

6. While defining a function which of the following symbol is used.

(a) ; (semicolon) (b) . (dot) **(c) : (colon)** (d) $ (dollar)

7. In which arguments the correct positional order is passed to a function?

**(a) Required** (b) Keyword (c) Default (d) Variable-length

8. Read the following statement and choose the correct statement(s).

(I) In Python, you don't have to mention the specific data types while defining function.

(II) Python keywords can be used as function name.

**(a) I is correct and II is wrong**

(b) Both are correct

(c) I is wrong and II is correct

(d) Both are wrong

9. Pick the correct one to execute the given statement successfully.

if _____ : print(x, " is a leap year")

(a) x%2=0 **(b) x%4==0** (c) x/4=0 (d) x%4=0

10. Which of the following keyword is used to define the function testpython(): ?

(a) define (b) pass **(c) def** (d) while

## Section-B

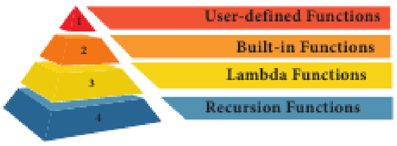
**Answer the following questions (2 Mark)**

**1. What is function?**

- Functions are named blocks of code that are designed to do one specific job.
- Function blocks begin with the keyword "def " followed by function name and parenthesis ().

**2. Write the different types of function.**

**TYPES OF FUNCTION:**



**3. What are the main advantages of function?**

- It avoids repetition and makes high degree of code reusing.
- It provides better modularity for your application.

**4. What is meant by scope of variable? Mention its types.**

- Scope of variable refers to the part of the program, where it is accessible, i.e., area where you can refer (use) it.
- Scope holds the current set of variables and their values.
- The two types of scopes are- **local scope** and **global scope**.

**5. Define global scope.**

- A variable, with global scope can be used anywhere in the program.
- It can be created by defining a variable outside the scope of any function/block.

## Section-C

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

**1. Write the rules of local variable.**

- A variable with local scope can be accessed only within the function/block that it is created in.

- When a variable is created inside the function/block, the variable becomes local to it.
- A local variable only exists while the function is executing.

**2. Write the basic rules for global keyword in python.**

- When we define a variable outside a function, it’s global by default.
- We use global keyword to read and write a global variable inside a function.
- Use of global keyword outside a function has no effect.

**3. What happens when we modify global variable inside the function?**

- If we modify the global variable , We can see the change on the **global** variable outside the function also.

**Example:**

```
x = 0
def add():
        global x
        x = x + 5

     print ("Inside add() function x value is :", x)
add()
print ("In main x value is :", x)
```

**5. Write a Python code to check whether a given year is leap year or not.**

**CODE:**

```
n=int(input("Enter the year"))
if(n%4==0):
   print ("Leap Year")
else:
   print ("Not a Leap Year")
```

**Output:**

```
Enter the year        2012
Leap Year
```

**8. What are the points to be noted while defining a function?**

- Function blocks begin with the keyword **“def”** followed by function name and parenthesis ().
- Any input parameters should be placed within these parentheses.
- The code block always comes after a colon (:) and is indented.
- The statement **“return [expression]”** exits a function, and it is optional.

## Section - D

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

**1. Explain the different types of function with an example.**

- Functions are named blocks of code that are designed to do one specific job.
- **Types of Functions**
  - User defined Function
  - Built-in Function
  - Lambda Function
  - Recursion Function

**i) BUILT-IN FUNCTION:**

- Built-in functions are Functions that are inbuilt with in Python.
- print(), echo() are some built-in function.

**ii) USER DEFINED FUNCTION:**

- Functions defined by the users themselves are called user defined function.
- Functions must be defined, to create and use certain functionality.
- Function blocks begin with the keyword “def ” followed by function name and parenthesis ().

**iii) LAMBDA FUNCTION:**

- In Python, anonymous function is a function that is defined without a name.
- While normal functions are defined using the **def** keyword, in Python anonymous functions are defined using the **lambda** keyword.
- Hence, anonymous functions are also called as **lambda** functions.

**iv) RECURSIVE FUNCTION:**

A function that calls itself is known as recursive.

**Overview of 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.

**2. Explain the scope of variables with an example.**

- Scope of variable refers to the part of the program, where it is accessible, i.e., area where you can refer (use) it.

➢ **Local Scope:**

- A variable declared inside the function's body or in the local scope is known as local variable.

**Rules of local variable:**

- A variable with local scope can be accessed only within the function/block that it is created in.
- When a variable is created inside the function/block, the variable becomes local to it.

➢ **Global Scope**

- A variable, with global scope can be used anywhere in the program.
- It can be created by defining a variable outside the scope of any function/block.

## ➢ Rules of global Keyword

- When we define a variable outside a function, it's global by default. You don't have to use global keyword.
- We use global keyword to read and write a global variable inside a function.

**4. Write a Python code to find the L.C.M. of two numbers.**

**CODE:**

```
x=int(input("Enter first number:"))
y=int(input("Enter second number:"))
if x>y:
      min=x
else:
      min=y
while(1):
   if((min%x == 0) and (min % y == 0)):
         print("LCM is:",min)
         break
   min=min+1
```

**OUTPUT:**

```
Enter first number:2
Enter second number:3
LCM is: 6
```

# 8. STRINGS AND STRING MANIPULATION

## Section – A

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

1. Which of the following is the output of the following python code?

```
str1="TamilNadu"
print(str1[::-1])
```

(a) Tamilnadu (b) Tmlau (c) udanlimaT **d) udaNlimaT**

2. What will be the output of the following code?

```
str1 = "Chennai Schools"
str1[7] = "-"
```

(a) Chennai-Schools (b) Chenna-School **(c) Type error** (d) Chennai

3. Which of the following operator is used for concatenation?

**(a) +** (b) & (c) * (d) =

4. Defining strings within triple quotes allows creating:

(a) Single line Strings **(b) Multiline Strings**

(c) Double line Strings (d) Multiple Strings

5. Strings in python:

(a) Changeable (b) Mutable **(c) Immutable** (d) flexible

6. Which of the following is the slicing operator?

(a) { } **(b) [ ]** (c) < > (d) ( )

7. What is stride?

(a) index value of slide operation (b) first argument of slice operation

(c) second argument of slice operation **(d) third argument of slice operation**

10. The subscript of a string may be:

(a) Positive (b) Negative (c) Both (a) and (b) **(d) Either (a) or (b)**

## Section-B

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

**1. What is String?**

- String is a data type in python, used to handle array of characters.
- String is a sequence of characters that may be a combination of letters, numbers, or special symbols enclosed within single, double or even triple quotes.

**2. Do you modify a string in Python?**

- No we cannot modify the string in python.
- String is an immutable

**3. How will you delete a string in Python?**

- Python will not allow deleting a particular character in a string.
- Whereas you can remove entire string variable using **del** command.

**4. What will be the output of the following python code?**

```
str1 = "School"
print(str1*3)
```

**OUTPUT:**

```
School School School
```

**5. What is slicing?**

- Slice is a substring of a main string.
- A substring can be taken from the original string by using [ ] slicing operator and index or subscript values.

## Section-C

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

**1. Write a Python program to display the given pattern**

```
C O M P U T E R
C O M P U T E
C O M P U T
C O M P U
C O M P
```

```
C O M
C O
C
```

**CODE:**

```
str="COMPUTER"
index=len(str)
for i in str:
        print(str[:index])
        index-=1
```

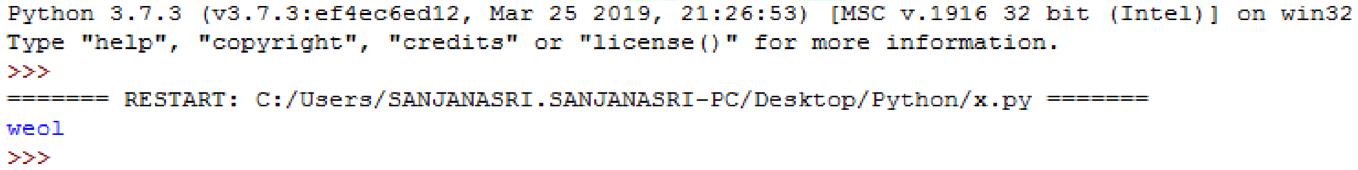
**3. What will be the output of the given python program?**

**CODE:**

```
str1 = "welcome"

str2 = "to school"

str3=str1[:2]+str2[len(str2)-2:]

print(str3)
```

**OUTPUT:**

**weol**



```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======= RESTART: C:/Users/SANJANASRI.SANJANASRI-PC/Desktop/Python/x.py =======
weol
>>>
```

**4. What is the use of format( )? Give an example.**

- The **format( )** function used with strings is very powerful function used for formatting strings.
- The curly braces { } are used as placeholders or replacement fields which get replaced along with format( ) function.

**EXAMPLE:**

```
num1=int (input("Number 1: "))
num2=int (input("Number 2: "))
print ("The sum of { } and { } is { }".format(num1, num2,(num1+num2)))
```

## Section - D

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

**1. 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**.

### (ii) Append (+ =)

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

### (iii) Repeating (*)

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

### (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.

**General format of slice operation:**

```
str[start:end]
```

### (v) Stride when slicing string

- When the slicing operation, you can specify a third argument as the stride
- 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.

# 9. LISTS, TUPLES, SETS, AND DICTIONARY

## Section – A

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

1. Pick odd one in connection with collection data type

(a) List (b) Tuple (c) Dictionary **(d) Loop**

2. Let list1=[2,4,6,8,10], then print(List1[-2]) will result in

(a) 10 **(b) 8** (c) 4 (d) 6

3. Which of the following function is used to count the number of elements in a list?

(a) count() (b) find() **(c)len()** (d) index()

4. If List=[10,20,30,40,50] then List[2]=35 will result

(a) [35,10,20,30,40,50] (b) [10,20,30,40,50,35]

**(c) [10,20,35,40,50]** (d) [10,35,30,40,50]

5. If List=[17,23,41,10] then List.append(32) will result

(a) [32,17,23,41,10] **(b) [17,23,41,10,32]**

(c) [10,17,23,32,41] (d) [41,32,23,17,10]

6. Which of the following Python function can be used to add more than one element within an Existing list?

(a) append() (b) append_more() **(c)extend()** (d) more()

7. What will be the result of the following Python code?

```
S=[x**2 for x in range(5)]
print(S)
```

(a) [0,1,2,4,5] **(b) [0,1,4,9,16]** (c) [0,1,4,9,16,25] (d) [1,4,9,16,25]

8. What is the use of type() function in python?

(a) To create a Tuple  (b) To know the type of an element in tuple.

**(c) To know the data type of python object.**  (d) To create a list.

9. Which of the following statement is not correct?

(a) A list is mutable

(b) A tuple is immutable.

(c) The append() function is used to add an element.

**(d) The extend() function is used in tuple to add elements in a list.**

10. Let setA={3,6,9}, setB={1,3,9}. What will be the result of the following snippet?

```
print(setA|setB)
```

(a) {3,6,9,1,3,9}  (b) {3,9}  (c) {1}  **(d) {1,3,6,9}**

11. Which of the following set operation includes all the elements that are in two sets but not the one that are common to two sets?

**(a) Symmetric difference**  (b) Difference  (c) Intersection  (d) Union

## Section-B

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

**1. What is List in Python?**

- A list is an ordered collection of values enclosed within square brackets [ ] also known as a “sequence data type”.
- Each value of a list is called as element.

**2. How will you access the list elements in reverse order?**

- Python enables reverse or negative indexing for the list elements.
- A negative index can be used to access an element in reverse order.
- The python sets -1 as the index value for the last element in list and -2 for the preceding element and so on.

**3. What will be the value of x in following python code?**

```
List1=[2,4,6,[1,3,5]]

x=len(List1)

print(x)
```

**OUTPUT:**

```
====== RESTART: C:/Users/SANJANASRI.SANJANASRI-PC/Desktop/Python/LI.py ======
4
>>>
```

**4. Differentiate del with remove( ) function of List.**

| del | remove( ) |
|---|---|
| **del** statement is used to delete known elements | **remove( )** function is used to delete elements of a list if its index is unknown. |
| The del statement can also be used to delete entire list. | The remove is used to delete a particular element |

**5. Write the syntax of creating a Tuple with n number of elements.**

**Syntax:**

**Tuple_Name = (E1, E2, E2 ……. En)** # Tuple with n number elements
**Tuple_Name = E1, E2, E3 ….. En** # Elements of a tuple without parenthesis

**6. What is set in Python?**

- In python, a set is another type of collection data type.
- A Set is a mutable and an unordered collection of elements without duplicates or repeated element.
- This feature used to include membership testing and eliminating duplicate elements.

## Section-C

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

**1. What are the advantages of Tuples over a list?**

- The elements of a list are changeable (mutable) whereas the elements of a tuple are unchangeable (immutable), this is the key difference between tuples and list.
- The elements of a list are enclosed within square brackets. But, the elements of a tuple are enclosed by paranthesis.
- Iterating tuples is faster than list.

**3. What will be the output of the following code?**

```
list = [2**x for x in range(5)]

print(list)
```

**OUTPUT:** **[1, 2, 4, 8, 16]**

**5. List out the set operations supported by python.**

**(i) Union**
**(ii) Intersection**
**(iii) Difference**
**iv) Symmetric difference**

## Section - D

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

**1. What the different ways to insert an element in a list. Explain with suitable example.**

**Inserting elements in a list using insert():**

- The **insert ( )** function helps you to include an element at your desired position.
- The **insert( )** function is used to insert an element at any position of a list.

**Syntax:**

```
List.insert (position index, element)
```

**Adding more elements in a list using append():**

- The **append( )** function is used to add a single element in a list.
- But, it includes elements at the end of a list.

**Syntax:**

```
List.append (element to be added)
```

**Adding more elements in a list using extend():**

- The **extend( )** function is used to add more than one element to an existing list.
- In extend( ) function, multiple elements should be specified within square bracket as arguments of the function.

**Syntax:**

```
List.extend ( [elements to be added])
```

**2. What is the purpose of range( )? Explain with an example.**

**range():**

- The range( ) is a function used to generate a series of values in Python.
- Using range( ) function, you can create list with series of values.
- The range( ) function has three arguments.

**Syntax of range ( ) function:**

```
range (start value, end value, step value)
```

**Creating a list with series of values**

- Using the range( ) function, you can create a list with series of values.
- To convert the result of range( ) function into list, we need one more function called list( ).
- The list( ) function makes the result of range( ) as a list.

**Syntax:**

```
List_Varibale = list ( range ( ) )
```

**3. What is nested tuple? Explain with an example.**

**Tuple:**

- Tuples consists of a number of values separated by comma and enclosed within parentheses.
- Tuple is similar to list, values in a list can be changed but not in a tuple.

**Nested Tuples:**

- In Python, a tuple can be defined inside another tuple; called Nested tuple.
- In a nested tuple, each tuple is considered as an element.
- The for loop will be useful to access all the elements in a nested tuple.

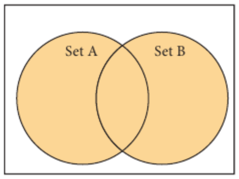
**4. Explain the different set operations supported by python with suitable example.**

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

**Set Operations:**

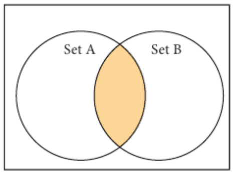
**(i) Union:**

- 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.



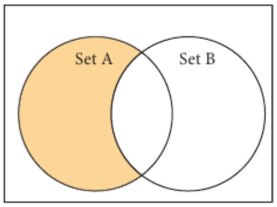
**(ii) Intersection:**

- 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.



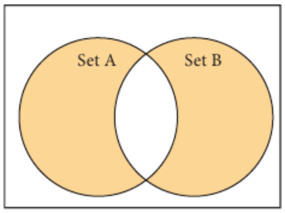
**(iii) Difference:**

- 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.



**(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.



# 10. PYTHON CLASSES AND OBJECTS

## Section – A

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

1. Which of the following are the key features of an Object Oriented Programming language?

(a) Constructor and Classes (b) Constructor and Object

**(c) Classes and Objects** (d) Constructor and Destructor

2. Functions defined inside a class:

(a) Functions (b) Module **(c) Methods** (d) section

3. Class members are accessed through which operator?

(a) & **(b) .** (c) # (d) %

4. Which of the following method is automatically executed when an object is created?

(a) __object__( ) (b) __del__( ) (c) __func__( ) **(d) __init__( )**

5. A private class variable is prefixed with

**(a) __** (b) && (c) ## (d) **

6. Which of the following method is used as destructor?

(a) __init__( ) (b) __dest__( ) (c) __rem__( ) **(d) __del__( )**

7. Which of the following class declaration is correct?

(a) class class_name (b) class class_name<> **(c) class class_name:** (d) class class_name[ ]

8. Which of the following is the output of the following program?

```
class Student:
def __init__(self, name):
self.name=name
S=Student("Tamil")
```

(a) Error **(b) Tamil** (c) name (d) self

9. Which of the following is the private class variable?

**(a) __num** (b) ##num (c) $$num (d) &&num

10. The process of creating an object is called as:

(a) Constructor (b) Destructor (c) Initialize **(d) Instantiation**

## Section-B

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

**1. What is class?**

- Class is the main building block in Python.
- Class is a template for the object.
- Object is a collection of data and function that act on those data.

**2. What is instantiation?**

- The process of creating object is called as “Class Instantiation”.

**3. What is the output of the following program?**

```
class Sample:
    __num=10
    def disp(self):
        print(self.__num)
S=Sample()
S.disp()
print(S.__num)
```

**OUTPUT:**

```
>>>
10
line 7, in <module>
    print(S.__num)
AttributeError: 'Sample' object has no attribute '__num'
```

**4. How will you create constructor in Python?**

- **“init”** is a special function begin and end with double underscore in Python act as a Constructor.
- Constructor function will automatically executed when an object of a class is created.

**5. What is the purpose of Destructor?**

- Destructor is also a special method gets executed automatically when an object exit from the scope.
- In Python, __del__( ) method is used as destructor.

## Section-C

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

**1. What are class members? How do you define it?**

- Variables defined inside a class are called as “Class Variable” and functions are called as “Methods”.
- Class variable and methods are together known as members of the class.
- The class members should be accessed through objects or instance of class.

**2. Write a class with two private class variables and print the sum using a method.**

**CODE:**

```
class Sample:
    def __init__(self,n1,n2):
        self.__n1=n1
        self.__n2=n2
    def sum(self):
        print("Class Variable 1:",self.__n1)
        print("Class Variable 2:",self.__n2)
        print("Sum:",self.__n1 + self.__n2)
S=Sample(5,10)
```

```
S.sum()
```

**OUTPUT:**

```
>>>
Class Variable 1: 5
Class Variable 2: 10
Sum: 15
>>>
```

**3. Find the error in the following program to get the given output?**

**ERROR CODE:**

```
class Fruits:
def __init__(self, f1, f2):
self.f1=f1
self.f2=f2
def display(self):
print("Fruit 1 = %s, Fruit 2 = %s" %(self.f1, self.f2))
F = Fruits ('Apple', 'Mango')
del F.display
F.display()
```

**OUTPUT:**

**Fruit 1 = Apple, Fruit 2 = Mango**

**ERROR:**

```
line 8, in <module>
   del F.display
AttributeError: display
```

**CORRECT CODE:**

```
class Fruits:
   def __init__(self, f1, f2):
      self.f1=f1
      self.f2=f2
   def display(self):
      print("Fruit 1 = %s, Fruit 2 = %s" %(self.f1, self.f2))
F = Fruits ('Apple','Mango')
F.display()
```

**OUTPUT:**

```
                  Fruit 1 = Apple, Fruit 2 = Mango
```

**4. What is the output of the following program?**

**CODE:**

```
class Greeting:
def __init__(self, name):
self.__name = name
def display(self):
print("Good Morning ", self.__name)

obj=Greeting('Bindu Madhavan')
obj.display()
```

**Output:**

```
>>>
Good Morning  Bindu Madhavan
>>>
```

**5. How do define constructor and destructor in Python?**

**CONSTRUCTOR:**

- **"init"** is a special function begin and end with double underscore in Python act as a Constructor.
- Constructor function will automatically executed when an object of a class is created.

**General format of constructor:**

```
def __init__(self, [args ……..]):
        <statements>
```

**DESTRUCTOR:**

- Destructor is also a special method gets executed automatically when an object exit from the scope.
- In Python, __del__( ) method is used as destructor.

**General format of destructor:**

```
def __del__(self):
        <statements>
```

## Section - D

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

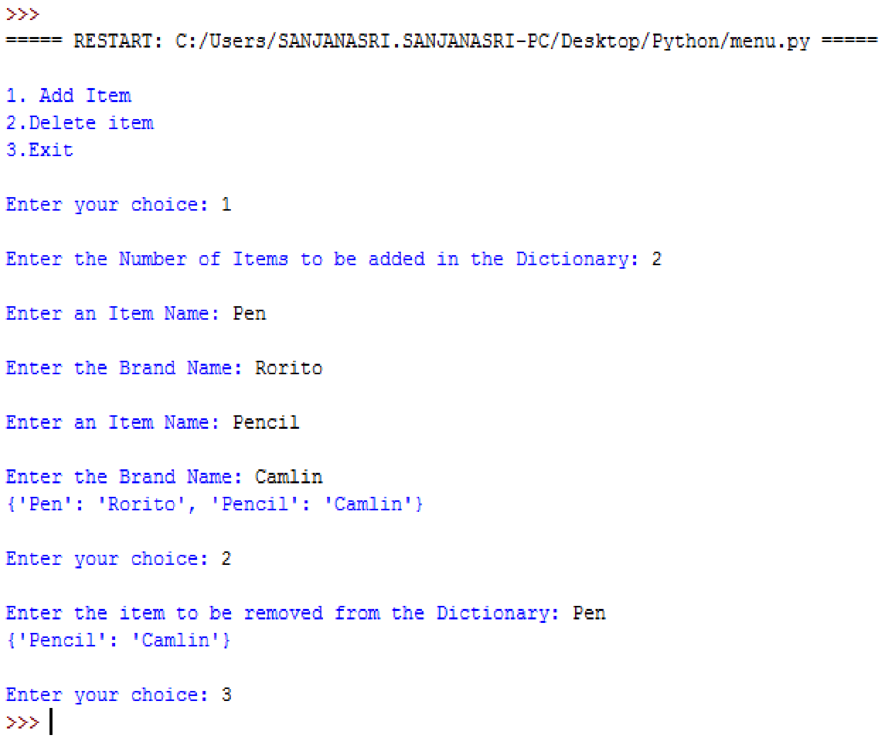
**1. Write a menu driven program to add or delete stationary items. You should use dictionary to store items and the brand.**

**CODE:**

```
stationary={}
print("\n1. Add Item \n2.Delete item \n3.Exit")
ch=int(input("\nEnter your choice: "))

while(ch==1)or(ch==2):
   if(ch==1):
      n=int(input("\nEnter the Number of Items to be added in the Dictionary: "))
      for i in range(n):
         item=input("\nEnter an Item Name: ")
```

```
            brand=input("\nEnter the Brand Name: ")
            stationary[item]=brand
        print(stationary)
    elif(ch==2):
        ritem=input("\nEnter the item to be removed from the Dictionary: ")
        stationary.pop(ritem)
        print(stationary)
    ch=int(input("\nEnter your choice: "))
```

**OUTPUT:**



```
>>>
===== RESTART: C:/Users/SANJANASRI.SANJANASRI-PC/Desktop/Python/menu.py =====

1. Add Item
2.Delete item
3.Exit

Enter your choice: 1

Enter the Number of Items to be added in the Dictionary: 2

Enter an Item Name: Pen

Enter the Brand Name: Rorito

Enter an Item Name: Pencil

Enter the Brand Name: Camlin
{'Pen': 'Rorito', 'Pencil': 'Camlin'}

Enter your choice: 2

Enter the item to be removed from the Dictionary: Pen
{'Pencil': 'Camlin'}

Enter your choice: 3
>>> |
```

## 11. DATABASE CONCEPTS

### Section – A

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

1. What is the acronym of DBMS?

a) DataBase Management Symbol b) Database Managing System

**c) DataBase Management System** d) DataBasic Management System

2. A table is known as

a) tuple b) attribute **c) relation** d)entity

9. A tuple is also known as

a) table **b) row** c) attribute d) field

## Section-B

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

**1. Mention few examples of a database.**

- Foxpro
- dbase.
- IBM DB2.
- Microsoft Access.
- Microsoft Excel.

**3. What is data consistency?**

- Data Consistency means that data values are the same at all instances of a database.

## Section - D

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

**5. Explain the characteristics of DBMS.**

| | |
|---|---|
| **1. Data Stored in a Tables** | • Data is stored into tables, created inside the database. |
| **2. Reduced Redundancy** | • Unnecessary repetition of data in database was a big problem. |
| **3.Data Consistency** | • Data Consistency means that data values are the same at all instances of a database. |
| **4.Support Multiple user and Concurrent Access** | • DBMS allows multiple users to work on it(update, insert, delete data) |
| **5.Query Language** | • DBMS provides users with a simple query language. |
| **6. Security** | • The DBMS also takes care of the security of data. |
| **7. DBMS Supports Transactions** | • It allows us to better handle and manage data integrity in real world applications |

# 12. STRUCTURED QUERY LANGUAGE

## Section – A

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

1. Which commands provide definitions for creating table structure, deleting relations, and modifying relation schemas.

**a. DDL** b. DML c. DCL d. DQL

2. Which command lets to change the structure of the table?

a. SELECT b. ORDER BY c. MODIFY **d. ALTER**

3. The command to delete a table is

**a. DROP** b. DELETE c. DELETE ALL d. ALTER TABLE

4. Queries can be generated using

**a. SELECT** b. ORDER BY c. MODIFY d. ALTER

5. The clause used to sort data in a database

a. SORT BY **b. ORDER BY** c. GROUP BY d. SELECT

## Section-B

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

**1. Write a query that selects all students whose age is less than 18 in order wise.**

**Query:** SELECT * FROM Student WHERE Age<=18 ORDER BY Name;

**2. Differentiate Unique and Primary Key constraint.**

| Unique Key Constraint | Primary Key Constraint |
|---|---|
| - This constraint ensures that no two rows have the same value in the specified columns. | - This constraint declares a field as a Primary key which helps to uniquely identify a record. |
| - The **UNIQUE** constraint can be applied only to fields that have also been declared as **NOT NULL**. | - The primary key does not allow **NULL** values and therefore a primary key field must have the **NOT NULL** constraint. |

**3. Write the difference between table constraint and column constraint?**

| Table Constraint | Column Constraint |
|---|---|
| - Table constraints apply to a group of one or more columns. | - Column constraints apply only to individual column. |

**4. Which component of SQL lets insert values in tables and which lets to create a table?**

| Command | Description | component |
|---|---|---|
| **Insert** | Inserts data into a table | DML |
| **Create** | To create tables in the database. | DDL |

**5. What is the difference between SQL and MySQL?**

| SQL | MySQL |
|---|---|
| - **Structured Query Language** is a language used for accessing databases. | - **MySQL** is a database management system, like SQL Server, Oracle, Informix, Postgres, etc. |
| - SQL is a **DBMS** | - **MySQL** is a **RDBMS**. |

## Section-C

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

**1. What is a constraint? Write short note on Primary key constraint.**

- Constraint is a condition applicable on a field or set of fields.
- Primary constraint declares a field as a Primary key which helps to uniquely identify a record.
- The primary key does not allow **NULL** values and therefore a primary key field must have the **NOT NULL** constraint.

**2. Write a SQL statement to modify the student table structure by adding a new field.**

Syntax : ***ALTER TABLE <table-name> ADD <column-name><data type><size>;***

To add a new column "Address" of type 'char' to the Student table, the command is used as

**Statement:** ALTER TABLE Student ADD Address char;

**3. Write any three DDL commands.**

**Data Definition Language:**

**Create Command:** To create tables in the database.

```
CREATE TABLE Student  (Admno integer,  Name char(20),  Gender char(1),  Age integer);
```

**Alter Command:** Alters the structure of the database.

```
ALTER TABLE Student ADD Address char;
```

**Drop Command:** Delete tables from database.

```
DROP TABLE Student;
```

**4. Write the use of Savepoint command with an example.**

- The **SAVEPOINT** command is used to temporarily save a transaction so that you can rollback to the point whenever required.

**Syntax:** SAVEPOINT savepoint_name;

**Example:** SAVEPOINT  A;

**5. Write a SQL statement using DISTINCT keyword.**

- The **DISTINCT** keyword is used along with the **SELECT** command to eliminate duplicate rows in the table.
- This helps to eliminate redundant data.
- **For Example:** SELECT DISTINCT Place FROM Student;

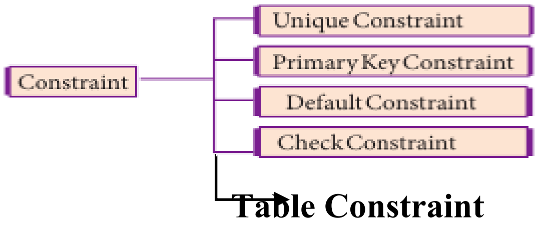
## Section - D

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

**1. Write the 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:

- 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:

### (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.

### (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.

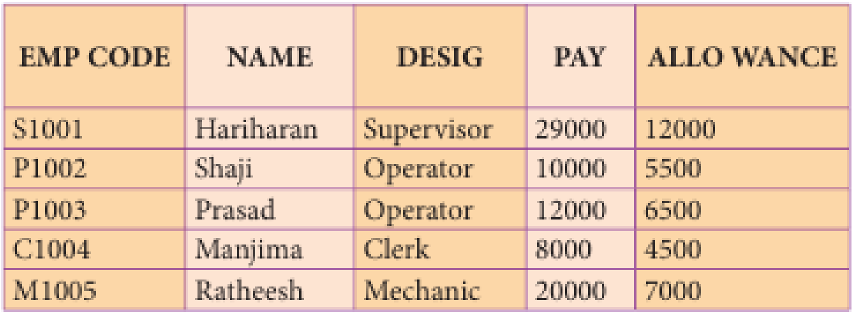
### (iv) Check Constraint:

- 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.

### (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.

**2. Consider the following employee table. Write SQL commands for the qtns.(i) to (v).**



| EMP CODE | NAME | DESIG | PAY | ALLO WANCE |
|---|---|---|---|---|
| S1001 | Hariharan | Supervisor | 29000 | 12000 |
| P1002 | Shaji | Operator | 10000 | 5500 |
| P1003 | Prasad | Operator | 12000 | 6500 |
| C1004 | Manjima | Clerk | 8000 | 4500 |
| M1005 | Ratheesh | Mechanic | 20000 | 7000 |

**(i) To display the details of all employees in descending order of pay.**

SELECT * FROM employee ORDER BY DESC;

**(ii) To display all employees whose allowance is between 5000 and 7000.**

SELECT * FROM employee WHERE allowance BETWEEN 5000 AND 7000;

**(iii) To remove the employees who are mechanic.**

DELETE FROM employee WHERE desig='Mechanic';

**(iv) To add a new row.**

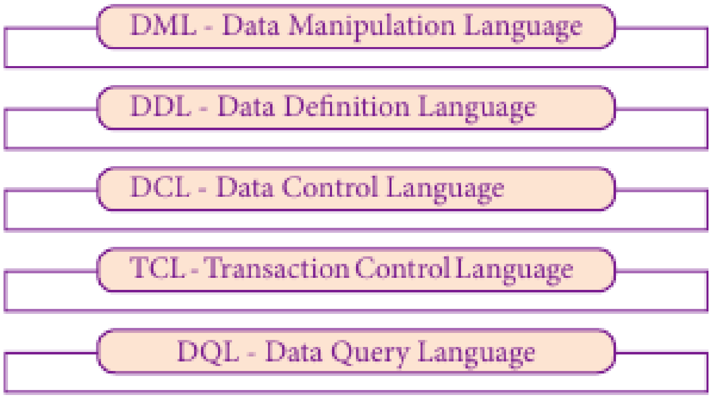
INSERT INTO employee
(empcode,name,desig,pay,allowance)VALUES(S1002,Baskaran,Supervisor,29000,12000);

**(v) To display the details of all employees who are operators.**

SELECT * FROM employee WHERE design='Operator';

**3. What are the components of SQL? Write the commands in each.**

**Components of SQL:**



DML - Data Manipulation Language

DDL - Data Definition Language

DCL - Data Control Language

TCL - Transaction Control Language

DQL - Data Query Language

## i) DATA MANIPULATION LANGUAGE :

- **A Data Manipulation Language (DML)** is a computer programming language used for adding (inserting), removing (deleting), and modifying (updating) data in a database.

## ii) DATA DEFINITION LANGUAGE:

- The **Data Definition Language (DDL)** consist of SQL statements used to define the database structure or schema.
- It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in databases.
- **SQL commands which comes under Data Definition Language are:**

| |
|---|
| **Create** |
| **Alter** |
| **Drop** |
| **Truncate** |

## iii) DATA CONTROL LANGUAGE:

- A **Data Control Language (DCL)** is a programming language used to control the access of data stored in a database.
- It is used for controlling privileges in the database (Authorization).

**SQL commands which come under Data Control Language are:**

| |
|---|
| **Grant** |
| **Revoke** |

## iv) TRANSACTIONAL CONTROL LANGUAGE:

- **Transactional control language (TCL)** commands are used to manage transactions in the database.

- These are used to manage the changes made to the data in a table by DML statements.

**SQL command which come under Transfer Control Language are:**

| |
|---|
| **Commit** |
| **Roll back** |
| **Save point** |

**4. Construct the following SQL statements in the student table:**

**(i) SELECT statement using GROUP BY clause.**

SELECT Gender FROM Student GROUP BY Gender;

**Output:**

| Gender |
|---|
| Male |
| Female |
| |

SELECT Gender, count(*) FROM Student GROUP BY male;

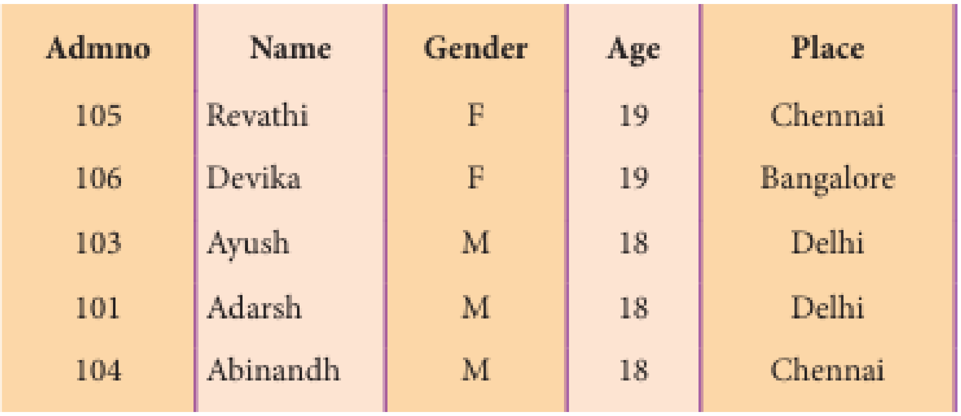
**Output:**

| Gender | Count(*) |
|---|---|
| Male | 5 |
| Female | 3 |

**(ii) SELECT statement using ORDER BY clause.**

SELECT * FROM student WHERE Age>=18 ORDER BY Name  DESC;

**Output:**



| Admno | Name | Gender | Age | Place |
|---|---|---|---|---|
| 105 | Revathi | F | 19 | Chennai |
| 106 | Devika | F | 19 | Bangalore |
| 103 | Ayush | M | 18 | Delhi |
| 101 | Adarsh | M | 18 | Delhi |
| 104 | Abinandh | M | 18 | Chennai |

**5. Write a SQL statement to create a table for employee having any five fields and create a table constraint for the employee table.**

```
CREATE TABLE employee
(
empno integer NOT NULL,
name char(20),
desig char(20),
pay integer,
allowance integer,
PRIMARY KEY (empno)
);
```

# 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<br>
mumba

(D) chennai,mylapore<br>
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<br>
Q<br>
H

**(D) Exam,**<br>
**Quarterly,**<br>
**Halfyearly**

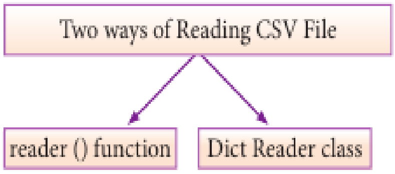
## Section-B

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

**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.

**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 Mark)**

**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. .

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

**PROGRAM:** student.csv

```
import csv
row = [‘3’, ‘Meena’,’Bangalore’]
with open(‘student.csv’, ‘r’) as readFile:
reader = csv.reader(readFile)
lines = list(reader)
lines[3] = row
with open(‘student.csv’, ‘w’) as writeFile:
writer = csv.writer(writeFile)
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
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']
```

**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 (,).

**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.

## Section - D

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

**1. Differentiate Excel file and CSV file.**

| Excel | CSV |
|---|---|
| • Excel is a binary file that holds information about all the worksheets in a file | • 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 | • 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. |
| 'x' | • Open a file for exclusive creation.. |
| 'a' | • Open for appending at the end of the file without truncating it. |
| 't' | • Opren in text mode. |
| 'b' | • Open in binary mode. |
| '+' | • Open a file for updating |

**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.**

   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.

**ii) CSV files- data with Spaces at the beginning**

- 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.

**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().

**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()
```

**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.

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

3. There may be an optional header line appearing as the first line of the file with the same format as normal record lines.

4. Within the header and each record, there may be one or more fields, separated by commas.

5. Each field may or may not be enclosed in double quotes.

6. Fields containing line breaks (CRLF), double quotes, and commas should be enclosed in double-quotes.

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

## 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 Mark)**

**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 . |

**2. Differentiate compiler and interpreter.**

| Compiler | Interpreter |
|---|---|
| Compiler generates an Intermediate Code. | Interpreter generates Machine Code. |
| Error deduction is difficult | Error deduction is easy |
| Comparatively faster | Slower |

**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.

**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.

## Section-C

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

**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 |
| • 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

**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]**
- **sys.argv[1]**
- **main(sys.argv[1])** :

## Section - D

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

**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.

### (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.

### (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])

**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**
  - ➢ **options**
  - ➢ **long_options**
  - ➢ **getopt()** method returns value consisting of two elements.
- Each of these values are stored separately in two different list (arrays) **opts and 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__':
 main(sys.argv[1:])
```

**OUTPUT:**

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

# 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 Mark)**

**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.

**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.

**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.

## Section-C

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

**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")
```

**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 Mark)**

**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.

**ADVANTAGES:**

- SQLite is fast, rigorously tested, and flexible, 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.

**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)
```

**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)
```

## 16. DATA VISUALIZATION USING PYPLOT: LINE CHART, PIE CHART AND BAR CHART

### Section – A

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

1. Which is a python package used for 2D graphics?

**a. matplotlib.pyplot** b. matplotlib.pip c. matplotlib.numpy d. matplotlib.plt

2. Identify the package manager for Python packages, or modules.

a. Matplotlib **b. PIP** c. plt.show() d. python package

3. Read the following code: Identify the purpose of this code and choose the right option from the following.

C:\Users\YourName\AppData\Local\Programs\Python\Python36-32\Scripts>pip – version

a. Check if PIP is Installed b. Install PIP c. Download a Package **d. Check PIP version**

4. Read the following code: Identify the purpose of this code and choose the right option from the following. C:\Users\Your Name\AppData\Local\Programs\Python\Python36-32\Scripts>pip list

**a. List installed packages** b. list command c. Install PIP d. packages installed

5. To install matplotlib, the following function will be typed in your command prompt.

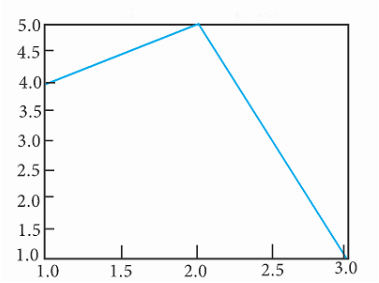
What does "-U"represents?

Python –m pip install –U pip

a. downloading pip to the latest version **b. upgrading pip to the latest version**

c. removing pip d. upgrading matplotlib to the latest version

6. Observe the output figure. Identify the coding for obtaining this output.



**a. import matplotlib.pyplot as plt**
**plt.plot([1,2,3],[4,5,1])**
**plt.show()**

b. import matplotlib.pyplot as plt
plt.plot([1,2],[4,5])
plt.show()

c. import matplotlib.pyplot as plt
plt.plot([2,3],[5,1])
plt.show()

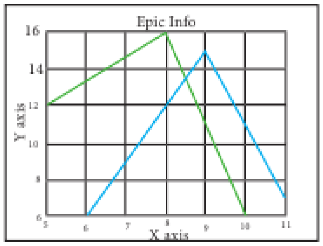
d. import matplotlib.pyplot as plt
plt.plot([1,3],[4,1])
plt.show()

7. Read the code:

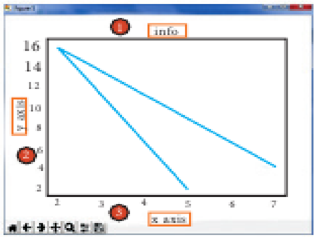
a. import matplotlib.pyplot as plt
b. plt.plot(3,2)
c. plt.show()

Identify the output for the above coding.

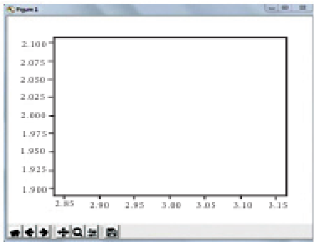
a.



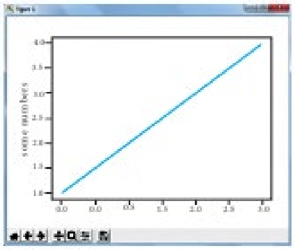
b.



[✓ c.]



d.



8. Which key is used to run the module?

a. F6　　b. F4　　c. F3　　**d. F5**

9. Identify the right type of chart using the following hints.

Hint 1: This chart is often used to visualize a trend in data over intervas of time.

Hint 2: The line in this type of chart is often drawn chronologically.

**a. Line chart**　　b. Bar chart　　c. Pie chart　　d. Scatter plot

10. Read the statements given below. Identify the right option from the following for pie chart.

Statement A: To make a pie chart with Matplotlib, we can use the plt.pie() function.

Statement B: The autopct parameter allows us to display the percentage value using the Python string formatting.

a. Statement A is correct　　b. Statement B is correct

**c. Both the statements are correct**　　d. Both the statements are wrong

## Section-B

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

**1. Define: Data Visualization.**

- Data Visualization is the graphical representation of information and data.
- The objective of Data Visualization is to communicate information visually to users using statistical graphics.

**2. List the general types of data visualization.**

- Charts
- Tables
- Graphs
- Maps
- Infographics
- Dashboards

**3. List the types of Visualizations in Matplotlib.**

- Line plot
- Scatter plot
- Histogram
- Box plot
- Bar chart and
- Pie chart

**4. How will you install Matplotlib?**

- **Matplotlib** can be installed using pip software.
- Pip is a management software for installing python packages.
- Importing Matplotlib using the command: **import matplotlib.pyplot as plt**
- Matplotlib can be imported in the workspace.

**5. Write the difference between the following functions:**
**plt.plot([1,2,3,4]), plt.plot([1,2,3,4], [1,4,9,16]).**

| **plt.plot([1,2,3,4])** | **plt.plot([1,2,3,4], [1,4,9,16])** |
|---|---|
| It refers y value as [1,2,3,4] | It refers x and y values as ([1,2,3,4], [1,4,9,16]) |
| Indirectly it refers x values as [0,1,2,3]<br>(0,1) (1,1) (2,3) (3,4) | Directly x and y values are given as<br>(1,1) (2,4) (3,9) (4,16) |

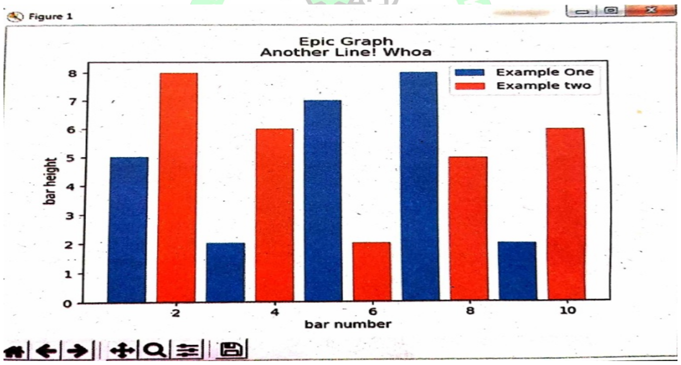
## Section-C

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

**1. Draw the output for the following data visualization plot.**

```
import matplotlib.pyplot as plt
plt.bar([1,3,5,7,9],[5,2,7,8,2], label="Example one")
plt.bar([2,4,6,8,10],[8,6,2,5,6], label="Example two", color='g')
plt.legend()
plt.xlabel('bar number')
plt.ylabel('bar height')
plt.title('Epic Graph\nAnother Line! Whoa')
plt.show()
```

**OUTPUT:**



**2. Write any three uses of data visualization.**

- Data Visualization help users to analyze and interpret the data easily.
- It makes complex data understandable and usable.
- Various Charts in Data Visualization helps to show relationship in the data for one or more variables.

**3. Write the coding for the following:**

**a. To check if PIP is Installed in your PC.**

- In command prompt type pip – version.
- If it is installed already, you will get version.
- **Command:** Python - m pip install - U pip

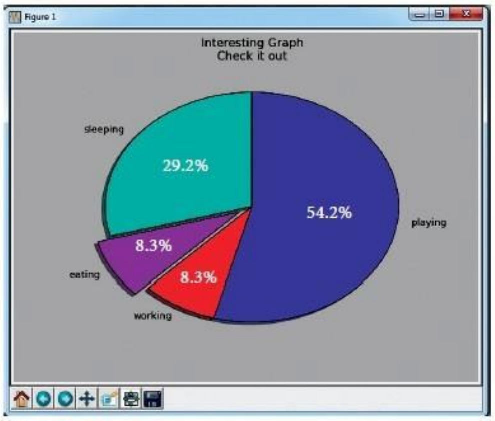
**b. To Check the version of PIP installed in your PC.**

- C:\Users\YourName\AppData\Local\Programs\Python\Python36-32\Scripts> pip-version

**c. To list the packages in matplotlib.**

- C:\Users\YourName\AppData\Local\Programs\Python\Python36-32\Scripts> pip list

**4. Write the plot for the following pie chart output.**



**Program:**

```
import matplotlib.pyplot as plt
slices=[7,2,2,13]
activities=['sleeping','eating', 'working','playing']
cols=['c','m','r','b']
plt.pie(slices, labels=activities, colors=cols,startangle=90, shadow=True,
explode=(0,0,0.1,0),autopct='%1.1f%%')
plt.title('Interesting Graph \nCheck it out')
plt.show()
```

**Calculation for the slices:**

29.2

100 x 24 = 7 [ since 24 hours a day]

8.3

100 x 24 = 1.99 =2

54.2

100 x 24 = 13 so the slices be [7,2,2,13]

## Section - D

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

**1. Explain in detail the types of pyplots using Matplotlib.**

**Line Chart:**

- A Line Chart or Line Graph is a type of chart which displays information as a series of data points called 'markers' connected by straight line segments.
- A Line Chart is often used to visualize a trend in data over intervals of time – a time series – thus the line is often drawn chronologically.

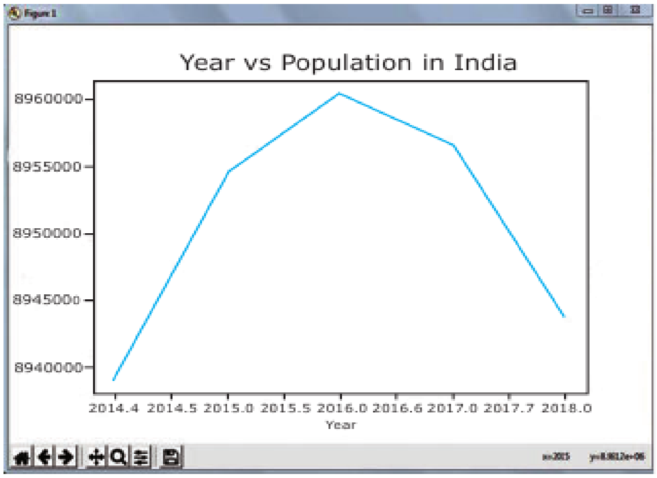
**In this program,**

Plt.title() → specifies title to the graph

Plt.xlabel() → specifies label for X-axis

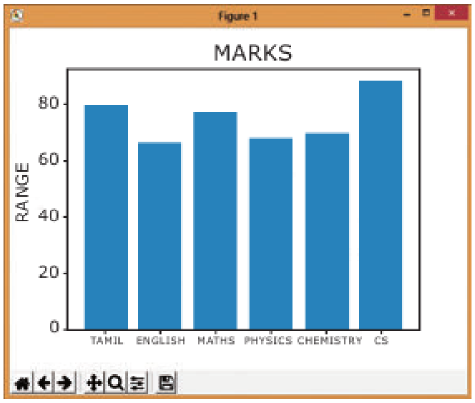
Plt.ylabel() → specifies label for Y-axis

## Output:



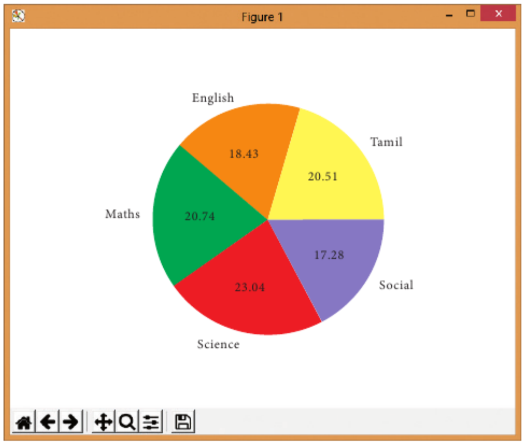
## Bar Chart:

- A BarPlot (or BarChart) is one of the most common type of plot.
- It shows the relationship between a numerical variable and a categorical variable.
- Bar chart represents categorical data with rectangular bars.
- Each bar has a height corresponds to the value it represents.
- The bars can be plotted vertically or horizontally.



## Pie Chart:

- Pie Chart is probably one of the most common type of chart.
- It is a circular graphic which is divided into slices to illustrate numerical proportion.
- The point of a pie chart is to show the relationship of parts out of a whole. .



**2. Explain the various buttons in a matplotlib window.**

**Home Button:**

- The Home Button will help once you have begun navigating your chart.
- If you ever want to return back to the original view, you can click on this.

**Forward/Back Buttons:**

- These buttons can be used like the Forward and Back buttons in your browser.
- You can click these to move back to the previous point you were at, or forward again.

**Pan Axis:**

- This cross-looking button allows you to click it, and then click and drag your graph around.

**Zoom:**

- The Zoom button lets you click on it, then click and drag a square that you would like to zoom into specifically.
- Zooming in will require a left click and drag.

**Configure Subplots:**

- This button allows you to configure various spacing options with your figure and plot.

**Save Figure:**

- This button will allow you to save your figure in various forms.

**3. Explain the purpose of the following functions:**

**a. plt.xlabel**

plt.xlabel()→specifies label for X-axis

**b. plt.ylabel**

plt.ylabel()→specifies label for Y-axis

**c. plt.title**

plt.title() →specifies title to the graph

**d. plt.legend()**

Calling legend() with no arguments automatically fetches the legend handles and their associated labels.

**e. plt.show()**

Display a figure. When running in Python with its Pylab mode,display all figures and return to the Python prompt.

**PREPARED BY**

**J. ILAKKIA** M.Sc., B.Ed., M.Phil.
**Computer Instructor Grade-I**
Govt. Hr. Sec. School
V.Pagandai, Villupuram 605 501.