

**2022-23  
EDITION**

# SURA'S 12<sup>TH</sup> STD SCHOOL GUIDES

SURA'S  
SUPER  
GUIDE

SURA'S

2022-23  
EDITION

## COMPUTER SCIENCE

Based on the updated new textbook



100%  
SUCCESS

- Exhaustive Additional MCQs, VSA, SA and LA questions with answers are given in each chapter.
- All the objective type (1 Mark) questions are given with 4 options.
  - (i) Choosing the correct option
  - (ii) Matching
  - (iii) Filling the blanks
  - (iv) Choosing the Correct/Incorrect Statement.
  - (v) Picking the Odd one Out.
  - (vi) Assertion and Reason.
- Model Question Papers 1 to 6 (PTA) : Questions are incorporated in the appropriate sections.
- Govt. Model Question Paper - 2019 (Govt. MQP-2019), Quarterly Exam - 2019 (QY-2019), Half Yearly Exam - 2019 (HY-2019), Public Exam March - 2020 (Mar-2020) and Govt. Supplementary Exam - Sep - 2020 (Sep-2020) are incorporated in the appropriate sections.
- Govt. Suppl. Exam 2020 question paper is given with answers.



12<sup>th</sup>  
STD.

Updated  
New Edition



**Call @**  
**9600175757**  
**8124301000**  
orders@surabooks.com

Buy Online @

  
**surabooks.com**



# COMPUTER SCIENCE

## 12<sup>th</sup> Standard

**FREE**  
Practice Workbook  
with  
Lab Manual

Based on the Updated New Textbook

### Salient Features

- Exhaustive Additional MCQs, VSA and SA question with answers are given in each chapter.
- All the objective type (1 Mark) questions are given with 4 options.
  - Choosing the correct option
  - Matching
  - Filling the blanks
  - Choosing the Correct/Incorrect Statement.
  - Picking the Odd one Out.
  - Assertion and Reason.
- Model Question Papers 1 to 6 (PTA) : Questions are incorporated in the appropriate sections.
- Govt. Model Question Paper - 2019 (Govt. MQP-2019), Quarterly Exam - 2019 (QY-2019), Half Yearly Exam - 2019 (HY-2019), Public Exam March - 2020 (Mar-2020) and Govt. Supplementary Exam - Sep - 2020 (Sep-2020) are incorporated in the appropriate sections.
- Govt. Suppl. Exam Sep - 2020 question paper is given with answers.



**SURA PUBLICATIONS**

Chennai

**2022-23 Edition**

**All rights reserved © SURA Publications.**

No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, digitally, electronically, mechanically, photocopying, recorded or otherwise, without the written permission of the publishers. Strict action will be taken.

**ISBN : 978-93-92559-13-6**

**Code No : SG91**

---

**Author :**

**Mr. Shanmugasundaram**

(Post Graduate Teacher, Chennai)

---

**Edited by :**

**Mrs. Malathy Krishnamoorthy M.Sc.,**

Erode

---

**Reviewed by :**

**Mr. Balaji M.Sc., M.Phil.**

Chennai

---

**Head Office:**

1620, 'J' Block, 16<sup>th</sup> Main Road,

Anna Nagar, Chennai - 600 040.

**Phone:** 044-4862 9977, 044-486 27755.

**Mob :** 81242 01000/ 81243 01000

**Fax :** (91) 44-26162173

**e-mail :** orders@surabooks.com

**website :** www.surabooks.com

**For More Information - Contact**

<b>Queries</b>	:	enquiry@surabooks.com
<b>For Order</b>	:	orders@surabooks.com
<b>Contact</b>	:	<b>96001 75757 / 8124301000</b>
<b>Whatsapp</b>	:	<b>8124201000 / 9840926027</b>
<b>Online Site</b>	:	<b>www.surabooks.com</b>
<b>For Free Study Materials Visit <a href="http://tnkalvi.in">http://tnkalvi.in</a></b>		

21-12-2021

## PREFACE

*“ The woods are lovely, dark and deep. “*

*But I have promises to keep, and  
miles to go before I sleep*

*- Robert Frost*

Respected Principals, Correspondents, Head Masters / Head Mistresses, Teachers,

From the bottom of our heart, we at SURA Publications sincerely thank you for the support and patronage that you have extended to us for more than a decade.

It is in our sincerest effort we take the pride of releasing **SURA’s Computer Science Guide** for +2 Standard – Edition 2022-23. This guide has been authored and edited by qualified teachers having teaching experience for over a decade in their respective subject fields. This Guide has been reviewed by reputed Professors who are currently serving as Head of the Department in esteemed Universities and Colleges.

With due respect to Teachers, I would like to mention that this guide will serve as a teaching companion to qualified teachers. Also, this guide will be an excellent learning companion to students with exhaustive exercises and in-text questions in addition to precise answers for textual questions.

In complete cognizance of the dedicated role of Teachers, I completely believe that our students will learn the subject effectively with this guide and prove their excellence in Board Examinations.

I once again sincerely thank the Teachers, Parents and Students for supporting and valuing our efforts.

God Bless all.

**Subash Raj, B.E., M.S.**

- Publisher

Sura Publications

*All the Best*

## CONTENTS

Unit	Chapter No	Title	Page No	Month
<b>UNIT- I</b> <b>Problem Solving Techniques</b>	1.	Function	1-8	June
	2.	Data Abstraction	9-17	June
	3.	Scoping	18-27	June
	4.	Algorithmic Strategies	28-43	June
<b>UNIT- II</b> <b>Core Python</b>	5.	Python -Variables and Operators	44-60	July
	6.	Control Structures	61-77	July
	7.	Python functions	78-98	July
	8.	Strings and String manipulations	99-114	Aug
<b>UNIT-III</b> <b>Modularity and OOPS</b>	9.	Lists, Tuples, Sets and Dictionary	115-138	Aug
	10.	Python Classes and objects	139-151	Aug
<b>UNIT-IV</b> <b>Database concepts and MySql</b>	11.	Database Concepts	152-169	Oct
	12.	Structured Query Language (SQL)	170-191	Oct
	13.	Python and CSV files	192-209	Oct
<b>UNIT-V</b> <b>Integrating Python with MySql and C++</b>	14.	Importing C++ programs in Python.	210-223	Oct
	15.	Data manipulation through SQL	224-236	Nov
	16.	Data visualization using pyplot: line chart, pie chart and bar chart	237-249	Nov
		Sep 2020 computer Science question paper with answers	251-264	

## UNIT-I PROBLEM SOLVING TECHNIQUES

### CHAPTER

# 1

# FUNCTION

### CHAPTER SNAPSHOT

- 1.1 Introduction
- 1.2 Function with respect to Programming language
  - 1.2.1 Function Specification
  - 1.2.2 Parameters (and arguments)
- 1.3 Interface Vs Implementation
  - 1.3.1 Characteristics of interface
- 1.4 Pure functions
  - 1.4.1 Impure functions
  - 1.4.2 Side-effects (Impure functions)
  - 1.4.3 Chameleons of Chromeland problem using function

### EVALUATION

#### PART - I

#### 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

[Ans. (a) Subroutines]
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

[Ans. (b) Function]
3. Which of the following is a distinct syntactic block?
  - (a) Subroutines
  - (b) Function
  - (c) Definition
  - (d) Modules

[Ans. (c) Definition]
4. The variables in a function definition are called as [PTA-2; QY-2019]
  - (a) Subroutines
  - (b) Function
  - (c) Definition
  - (d) Parameters

[Ans. (d) Parameters]
5. The values which are passed to a function definition are called [HY-2019]
  - (a) Arguments
  - (b) Subroutines
  - (c) Function
  - (d) Definition

[Ans. (a) Arguments]
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

[Ans. (b) Parentheses]
7. Which of the following defines what an object can do?
  - (a) Operating System
  - (b) Compiler
  - (c) Interface
  - (d) Interpreter

[Ans. (c) Interface]



**8. Which of the following carries out the instructions defined in the interface?**

- (a) Operating System (b) Compiler  
 (c) Implementation (d) Interpreter

[Ans. (c) Implementation]

**9. The functions which will give exact result when same arguments are passed are called**

[PTA-3; Mar.-2020]

- (a) Impure functions (b) Partial Functions  
 (c) Dynamic Functions (d) Pure functions

[Ans. (d) Pure functions]

**10. The functions which cause side effects to the arguments passed are called**

- (a) Impure function (b) Partial Functions  
 (c) Dynamic Functions (d) Pure functions

[Ans. (a) Impure function]

### PART - II

#### ANSWER THE FOLLOWING QUESTIONS

(2 MARKS)

**1. What is a subroutine?** [PTA-1; HY-2019]

**Ans. (i)** Subroutines are the basic building blocks of computer programs. Subroutines are small sections of code that are used to perform a particular task that can be used repeatedly.

**(ii)** In Programming languages these subroutines are called as Functions.

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

**Ans.** A function is a unit of code that is often defined within a greater code structure. Specifically, a function contains a set of code that works on many kinds of inputs, like variants, expressions and produces a concrete output.

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

**Ans.** X:= (78) has an expression in it but (78) is not itself an expression. Rather, it is a function definition. Definitions bind values to names, in this case the value 78 being bound to the name 'X'.

**4. Differentiate interface and implementation.**

**Ans.** The difference between interface and implementation is

Interface	Implementation
Interface just defines what an object can do, but won't actually do it	Implementation carries out the instructions defined in the interface

**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`  
 ii) `let disp :  
 print 'welcome'`  
 iii) `let rec sum num :  
 if (num!=0) then return num + sum  
 (num-1)  
 else  
 return num`

**Ans. (i)** Recursive function

**(ii)** Normal function

**(iii)** Recursive function

### PART - III

#### ANSWER THE FOLLOWING QUESTIONS

(3 MARKS)

**1. Mention the characteristics of Interface.**

[Sep-2020]

**Ans. (i)** The class template specifies the interfaces to enable an object to be created and operated properly.

**(ii)** An object's attributes and behaviour is controlled by sending functions to the object.

**2. Why strlen is called pure function?**

[Govt. MQP-2019]

**Ans. (i)** strlen is a pure function because the function takes one variable as a parameter, and accesses it to find its length.

**(ii)** This function reads external memory but does not change it, and the value returned derives from the external memory accessed.

**3. What is the side effect of impure function. Give example.**

[PTA-5]

**Ans.** Impure Function has the following side effects

**(i)** Function impure (has side effect) is that it doesn't take any arguments and it doesn't return any value.

**(ii)** Function depends on variables or functions outside of its definition block.

**(iii)** It never assure you that the function will behave the same every time it's called.

**For example :**

```
let y := 0
(int) inc (int) x
y: = y + x;
return (y)
```

- (iv) Here, the result of inc() will change every time if the value of 'y' get changed inside the function definition.
- (v) Hence, the side effect of inc () function is changing the data of the external variable 'y'.

**4. Differentiate pure and impure function.**

**Ans.** [PTA-3, 6; Mar.-2020]

S. No.	Pure	Impure
(i)	The return value of the pure functions solely depends on its arguments passed.	The return value of the impure functions does not solely depend on its arguments passed.
(ii)	If you call the pure functions with the same set of arguments, you will always get the same return values.	If you call the impure functions with the same set of arguments, you might get the different return values.
(iii)	They do not have any side effects.	They have side effects. For example, random(), Date().
(iv)	They do not modify the arguments which are passed to them	They may modify the arguments which are passed to them

**5. What happens if you modify a variable outside the function? Give an example.**

**Ans.** One of the most popular groups of side effects is modifying the variable outside of function.

**For example :**

```
let y: = 0
(int) inc (int) x
y: = y + x;
return (y)
```

Here, the result of inc () will change every time if the value of 'y' get changed inside the function definition. Hence, the side effect of inc () function is changing the data of the external variable 'y'.

**PART - IV**

**ANSWER THE FOLLOWING QUESTIONS**

**(5 MARKS)**

- 1. What are called Parameters and write a note on** [PTA-2]
- (i) Parameter without Type
  - (ii) Parameter with Type

**Ans. Parameters (and arguments) :** Parameters are the variables in a function definition and arguments are the values which are passed to a function definition.

- (i) **Parameter without Type :** Let us see an example of a function, definition :

(requires:  $b \geq 0$ )  
(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 which is passed to the variable 'b' is the argument. The precondition (**requires**) and postcondition (**returns**) of the function is given.
- Note we have not mentioned any types: (**data types**). Some language compiler solves this type (**data type**) inference problem algorithmically, but some require the type to be mentioned.
- In the above function definition if expression can return 1 in the then branch, by the **typing** rule the entire if expression has type **int**.
- Since the if expression has type '**int**', the function's return type also be '**int**'. '**b**' is compared to 0 with the equality operator, so '**b**' is also a type of '**int**'. Since 'a' is multiplied with another expression using the \* operator, '**a**' must be an int.

- (ii) **Parameter with Type :** Now let us write the same function definition with types for some reason:

```
(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)
```



- When we write the type annotations for 'a' and 'b' the parentheses are mandatory. Generally we can leave out these annotations, because it's simpler to let the compiler infer them.
- There are times we may want to explicitly write down types. This is useful on times when you get a type error from the compiler that doesn't make sense. Explicitly annotating the types can help with debugging such an error message.

2. Identify in the following program [PTA-5]

```
let rec gcd a b :=  
if b <> 0 then gcd b (a mod b) else return a
```

- Name of the function
- Identify the statement which tells it is a recursive function
- Name of the argument variable
- Statement which invoke the function recursively
- Statement which terminates the recursion

- Ans. (i) gcd  
(ii) let rec gcd  
(iii) a, b  
(iv) gcd b (a mod b)  
(v) return a

3. Explain with example Pure and impure functions.

Ans. Pure functions :

- Pure functions are functions which will give exact result when the same arguments are passed.
- For example the mathematical function sin (0) always results 0. This means that every time you call the function with the same arguments, you will always get the same result.
- A function can be a pure function provided it should not have any external variable which will alter the behaviour of that variable.

Let us see an example  
let square x  
return: x \* x

(iv) The above function square is a pure function because it will not give different results for same input.

(v) There are various theoretical advantages of having pure functions. One advantage is that if a function is pure, then if it is called several times with the same arguments, the compiler only needs to actually call the function once. Let's see an example

```
let i = 0;  
if i < strlen (s) then  
-- Do something which doesn't affect s  
++i
```

(vi) If it is compiled, strlen (s) is called each time and strlen needs to iterate over the whole of 's'. If the compiler is smart enough to work out that strlen is a pure function and that 's' is not updated in the loop, then it can remove the redundant extra calls to strlen and make the loop to execute only one time.

(vii) From these what we can understand, strlen is a pure function because the function takes one variable as a parameter, and accesses it to find its length. This function reads external memory but does not change it, and the value returned derives from the external memory accessed.

Impure functions :

(i) The variables used inside the function may cause side effects though the functions which are not passed with any arguments. In such cases the function is called impure function.

(ii) When a function depends on variables or functions outside of its definition block, you can never be sure that the function will behave the same every time it's called. For example the mathematical function random() will give different outputs for the same function call.

```
let Random number  
let a := random()  
if a > 10 then  
return: a  
else  
return: 10
```

(iii) Here the function Random is impure as it is not sure what will be the result when we call the function.

4. Explain with an example interface and implementation.

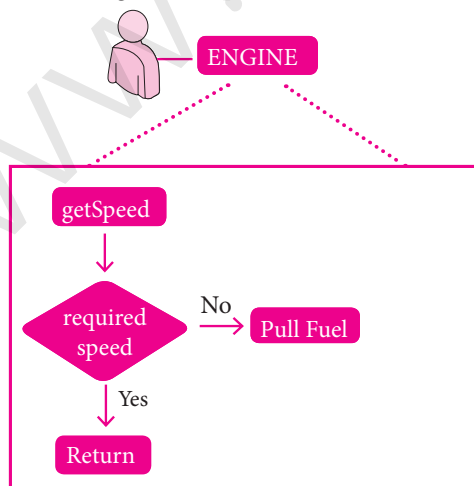
Ans. **Interface :**

- (i) An interface is a set of action that an object can do. For example when you press a light switch, the light goes on, you may not have cared how it splashed the light. In Object Oriented Programming language, an Interface is a description of all functions that a class must have in order to be a new interface.
- (ii) In our example, anything that **"ACTS LIKE"** a light, should have function definitions like turn\_on () and a turn\_off (). The purpose of interfaces is to allow the computer to enforce the properties of the class of **TYPE T** (whatever the interface is) must have functions called X, Y, Z, etc.
- (iii) A class declaration combines the external interface (its local state) with an implementation of that interface (the code that carries out the behaviour). An object is an instance created from the class. The interface defines an object's visibility to the outside world.

**Implementation :**

- (i) Implementation carries out the instructions defined in the interface.
- (ii) How the object is processed and executed is the implementation.
- (iii) A class declaration combines the external interface (its local state) with an implementation of that interface (the code that carries out the behaviour).

For example, let's take the example of increasing a car's speed.



- (iv) The person who drives the car doesn't care about the internal working. To increase the speed of the car he just presses the accelerator to get the desired behaviour. Here the accelerator is the interface between the driver (the calling / invoking object) and the engine (the called object).
- (v) In this case, the function call would be Speed (70): This is the interface. Internally, the engine of the car is doing all the things. It's where fuel, air, pressure, and electricity come together to create the power to move the vehicle.
- (vi) All of these actions are separated from the driver, who just wants to go faster. Thus we separate interface from implementation.

## HANDS ON PRACTICE

1. Write algorithmic function definition to find the minimum among 3 numbers.

Ans. let min 3 x y z :=  
 if x < y then  
     if x < z then x else z  
 else  
     if y < z then y else z

2. Write algorithmic recursive function definition to find the sum of n natural numbers.

Ans. let rec sum num:  
 if (num!=0) then return num+sum num-1)  
 else  
 return num

## PTA QUESTIONS AND ANSWERS

### 1 MARK

1. A function definition which call itself : [PTA-1]  
 (a) Pure function (b) Impure function  
 (c) Normal function  
 (d) Recursive function

[Ans. (d) Recursive function]

### 3 MARKS

1. Write a function that finds the minimum of its three arguments. [PTA-4; QY-2019]

Ans. let min 3 x y z :=  
 if x < y then  
     if x < z then x else z  
 else  
     if y < z then y else z

## CHAPTER SNAPSHOT

- 2.1 Data Abstraction – Introduction
- 2.2 Abstract Data Types
- 2.3 Constructors and Selectors
- 2.4 Representation of Abstract datatype using Rational numbers
- 2.5 Lists, Tuples
  - 2.5.1 List
  - 2.5.2 Tuple
- 2.6 Data Abstraction in Structure

## EVALUATION

### PART - I

#### CHOOSE THE BEST ANSWER (1 MARK)

1. Which of the following functions that build the abstract data type ? *[Sep-2020]*
  - (a) Constructors
  - (b) Destructors
  - (c) Recursive
  - (d) Nested

**[Ans. (a) Constructors]**
2. Which of the following functions that retrieve information from the data type?
  - (a) Constructors
  - (b) Selectors
  - (c) Recursive
  - (d) Nested

**[Ans. (b) Selectors]**
3. The data structure which is a mutable ordered sequence of elements is called
  - (a) Built in
  - (b) List
  - (c) Tuple
  - (d) Derived data

**[Ans. (b) List]**
4. A sequence of immutable objects is called *[Mar.-2020]*
  - (a) Built in
  - (b) List
  - (c) Tuple
  - (d) Derived data

**[Ans. (c) Tuple]**
5. The data type whose representation is known are called *[PTA-2; QY-2019]*
  - (a) Built in datatype
  - (b) Derived datatype
  - (c) Concrete datatype
  - (d) Abstract datatype

**[Ans. (c) Concrete datatype]**
6. The data type whose representation is unknown are called
  - (a) Built in datatype
  - (b) Derived datatype
  - (c) Concrete datatype
  - (d) Abstract datatype

**[Ans. (d) Abstract datatype]**
7. Which of the following is a compound structure?
  - (a) Pair
  - (b) Triplet
  - (c) Single
  - (d) Quadrat

**[Ans. (a) Pair]**

8. Bundling two values together into one can be considered as [Govt. MQP - 2019; PTA-4]

- (a) Pair (b) Triplet  
(c) Single (d) Quadrat

[Ans. (a) Pair]

9. Which of the following allow to name the various parts of a multi-item object? [PTA-6]

- (a) Tuples (b) Lists  
(c) Classes (d) Quadrats

[Ans. (c) Classes]

10. Which of the following is constructed by placing expressions within square brackets?

- (a) Tuples (b) Lists  
(c) Classes (d) Quadrats

[Ans. (b) Lists]

### PART - II

#### ANSWER THE FOLLOWING QUESTIONS

(2 MARKS)

1. What is abstract data type?

Ans. (i) 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.

(ii) The definition of ADT only mentions what operations are to be performed but not how these operations will be implemented.

2. Differentiate constructors and selectors.

Ans. [PTA-2, 3; QY-2019]

S. No.	Constructors	Selectors
(i)	Constructors are functions that build the abstract data type.	Selectors are functions that retrieve information from the data type.
(ii)	Constructors create an object, bundling together different pieces of information.	Selectors extract individual pieces of information from the object

3. What is a Pair? Give an example. [Mar.-2020]

Ans. (i) Any way of bundling two values together into one can be considered as a Pair. Lists are a common method to do so. Therefore List can be called as Pairs.

(ii) Example : List = [(10,10), (1,20)]

4. What is a List? Give an example. [QY - 2019]

Ans. (i) List is constructed by placing expressions within square brackets separated by commas.

(ii) Such an expression is called a list literal. List can store multiple values. Each value can be of any type and can even be another list.

Example : lst := [10, 20]  
x, y := lst

5. What is a Tuple? Give an example.

Ans. (i) A tuple is a comma-separated sequence of values surrounded with parentheses. Tuple is similar to a list.

(ii) The difference between the two is that you cannot change the elements of a tuple once it is assigned whereas in a list, elements can be changed.

(iii) Example : colour= ('red', 'blue', 'Green')

### PART - III

#### ANSWER THE FOLLOWING QUESTIONS

(3 MARKS)

1. Differentiate Concrete datatype and Abstract datatype.

Ans.

S. No.	Concrete datatype	Abstract datatype
(i)	Concrete datatypes or structures (CDT's) are direct implementations of a relatively simple concept.	Abstract Datatypes (ADT's) offer a high level view (and use) of a concept independent of its implementation.
(ii)	A concrete data type is a data type whose representation is known.	Abstract data type the representation of a data type is unknown.

2. Which strategy is used for program designing? Define that Strategy. [Govt. MQP-2019]

Ans. A powerful strategy for designing programs: 'wishful thinking'. Wishful Thinking is the formation of beliefs and making decisions according to what might be pleasing to imagine instead of by appealing to reality.

3. Identify Which of the following are constructors and selectors? [PTA-5]

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

- Ans. (a) Constructors  
 (b) Selectors  
 (c) Selectors  
 (d) Selectors  
 (e) Constructors  
 (f) Selectors

4. What are the different ways to access the elements of a list. Give example.

Ans. (i) The elements of a list can be accessed in two ways. The first way is via our familiar method of multiple assignment, which unpacks a list into its elements and binds each element to a different name.

```
lst := [10, 20]
x, y := lst
```

- (ii) In the above example x will become 10 and y will become 20.
- (iii) A second method for accessing the elements in a list is by the element selection operator, also expressed using square brackets. Unlike a list literal, a square-brackets expression directly following another expression does not evaluate to a list value, but instead selects an element from the value of the preceding expression.

```
lst[0]
10
lst[1]
20
```

5. Identify Which of the following are List, Tuple and class ?

- (a) arr [1, 2, 34]
- (b) arr (1, 2, 34)
- (c) student [rno, name, mark]
- (d) day= ('sun', 'mon', 'tue', 'wed')
- (e) x= [2, 5, 6.5, [5, 6], 8.2]
- (f) employee [eno, ename, esal, eaddress]

- Ans. (a) List  
 (b) Tuple  
 (c) Class  
 (d) Tuple  
 (e) List  
 (f) Class

### PART - IV

#### ANSWER THE FOLLOWING QUESTIONS

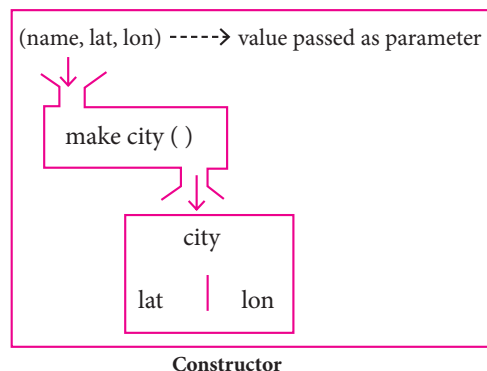
(5 MARKS)

1. How will you facilitate data abstraction. Explain it with suitable example. [PTA-2, 4]

Ans. 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**

#### Constructors :

- (i) Constructors are functions that build the abstract data type.
- (ii) Constructors create an object, bundling together different pieces of information.
- (iii) For example, say you have an abstract data type called city.
- (iv) This city object will hold the city's name, and its latitude and longitude.
- (v) To create a city object, you'd use a function like **city = makecity (name, lat, lon)**.
- (vi) Here makecity (name, lat, lon) is the constructor which creates the object city.

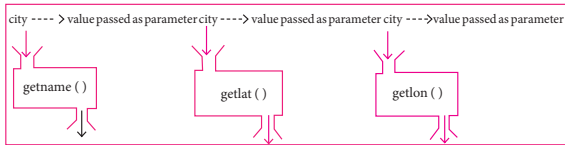


#### Selectors :

- (i) Selectors are functions that retrieve information from the data type.
- (ii) Selectors extract individual pieces of information from the object.



- (iii) To extract the information of a city object, you would use functions like  
 getname(city)  
 getlat(city)  
 getlon(city)  
 These are the selectors because these functions extract the information of the city object.



**2. What is a List? Why List can be called as Pairs. Explain with suitable example. [PTA-6]**

**Ans. List :**

- (i) List is constructed by placing expressions within square brackets separated by commas. Such an expression is called a list literal. List can store multiple values. Each value can be of any type and can even be another list.  
 Example for List is [10, 20].

- (ii) The elements of a list can be accessed in two ways. The first way is via our familiar method of multiple assignment, which unpacks a list into its elements and binds each element to a different name.

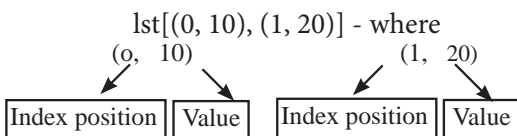
```
lst := [10, 20]
x, y := lst
```

- (iii) In the above example x will become 10 and y will become 20. A second method for accessing the elements in a list is by the element selection operator, also expressed using square brackets.

- (iv) Unlike a list literal, a square-brackets expression directly following another expression does not evaluate to a list value, but instead selects an element from the value of the preceding expression.

```
lst[0]
10
lst[1]
20
```

- (v) In both the example mentioned above mathematically we can represent list similar to a set.



**Pair :**

- (vi) Any way of bundling two values together into one can be considered as a pair. Lists are a common method to do so. Therefore List can be called as Pairs.

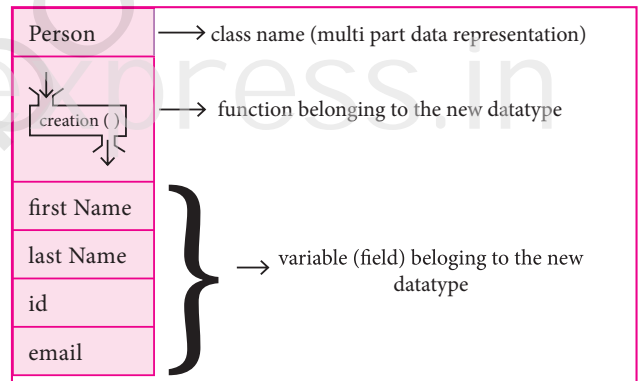
**3. How will you access the multi-item? Explain with example.**

**Ans. (i)** The structure construct (In OOP languages it's called class construct) is used to represent multi-part objects where each part is named (given a name). Consider the following pseudo code:

**class Person:**

```
creation()
firstName := ""
lastName := ""
id := ""
email := ""
```

The new data type Person is pictorially represented as



Let main() contains	
p1:=Person()	statement creates the object
firstName := "Padmashri"	setting a field called first Name with value Padmashri
lastName := "Baskar"	setting a field called lastName with value Baskar
id := "994-222-1234"	setting a field called id value 994-222-1234
email="compisci@gamil.com"	setting a filed called email with value compisci@gmail.com
- - output of firstName : Padmashri	



- (ii) The class (structure) construct defines the form for multi-part objects that represent a person.
- (iii) Person is referred to as a class or a type, while p1 is referred to as an object or an instance.
- (iv) Here class Person as a cookie cutter, and p1 as a particular cookie. Using the cookie cutter you can make many cookies. Same way using class created many objects of that type.
- (v) A class defines a data abstraction by grouping related data items. A class is not just data, it has functions defined within it. We say such functions are subordinate to the class because their job is to do things with the data of the class.

- (iii) getlon(city)  
are the selectors because these functions extract the information of the city object
- b) The two parts of a program are, the part that operates on abstract data and the part that defines a concrete representation, is connected by a small set of functions that implement abstract data in terms of the concrete representation.

### PTA QUESTIONS AND ANSWERS

#### 1 MARK

1. Expansion of ADT : [PTA-1]
  - (a) Abstract Data Tuple
  - (b) All Data Template
  - (c) Abstract Data Type
  - (d) Application Data Type
2. ADT can be implemented using \_\_\_\_\_. [PTA-5]
  - (a) singly linked list
  - (b) doubly linked list
  - (c) either A or B
  - (d) neither A nor B

[Ans. (c) Abstract Data Type]

[Ans. (a) singly linked list]

### GOVERNMENT EXAM QUESTIONS AND ANSWERS

#### 1 MARK

1. The datatype whose representation is unknown is called [HY-2019]
  - (a) Built-in datatype
  - (b) Derived datatype
  - (c) Concrete datatype
  - (d) Abstract datatype

[Ans. (d) Abstract datatype]

#### 3 MARKS

1. (a) What is selector? [Sep-2020]
  - (b) What are the parts of a program?
- Ans. a)** Selectors are nothing but the functions that retrieve information from the data type. Therefore in the above code

- (i) getname(city)
- (ii) getlat(city)

### ADDITIONAL QUESTIONS AND ANSWERS

#### CHOOSE THE CORRECT ANSWER 1 MARK

1. Which of the following is a powerful concept that allows programmers to treat codes as objects?

- (a) Encapsulation
- (b) Data Abstraction
- (c) Inheritance
- (d) Polymorphism

[Ans. (b) Data Abstraction]

2. Which of the following provides modularity?

- (a) Datatypes
- (b) Subroutines
- (c) Classes
- (d) Abstraction

[Ans. (d) Abstraction]

3. Which of the following is a type for objects whose behavior is defined by a set of value and a set of operations?

- (a) User-defined datatype
- (b) Derived datatype
- (c) Built-in datatype
- (d) Abstract datatype

[Ans. (d) Abstract datatype]

4. ADT behavior is defined by

- (i) Set of Variables
- (ii) Set of Value
- (iii) Set of Functions
- (iv) Set of Operations
- (a) i, ii
- (b) ii, iii
- (c) ii, iv
- (d) i, iii

[Ans. (c) ii, iv]

5. The process of providing only the essentials and hiding the details is known as

- (a) Functions
- (b) Abstraction
- (c) Encapsulation
- (d) Pairs

[Ans. (b) Abstraction]

6. Which of the following gives an implementation independent view?

- (a) Abstract
- (b) Concrete
- (c) Datatype
- (d) Behavior of an object

[Ans. (a) Abstract]

# CHAPTER 3

# SCOPING

## CHAPTER SNAPSHOT

- |                             |  |
|-----------------------------|--|
| 3.1 Introduction            | 3.5 Module   |
| 3.2 Variable Scope          | 3.5.1. Characteristics of Modules                        |
| 3.3 LEGB rule               | 3.5.2. The benefits of using modular programming include |
| 3.4 Types of Variable Scope | 3.5.3. Access Control                                    |
| 3.4.1. Local Scope          |  |
| 3.4.2. Global Scope         |  |
| 3.4.3. Enclosed Scope       |  |
| 3.4.4. Built-in-Scope       |  |

## EVALUATION

### PART - I

#### CHOOSE THE BEST ANSWER (1 MARK)

- 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  
[Ans. (a) Scope]
- The process of binding a variable name with an object is called [Sep-2020]  
(a) Scope (b) Mapping  
(c) late binding (d) early binding  
[Ans. (b) Mapping]
- Which of the following is used in programming languages to map the variable and object? [PTA-2; HY-2019]  
(a) :: (b) :=  
(c) = (d) ==  
[Ans. (c) =]
- Containers for mapping names of variables to objects is called [QY-2019]  
(a) Scope (b) Mapping  
(c) Binding (d) Namespaces  
[Ans. (d) Namespaces]
- Which scope refers to variables defined in current function?  
(a) Local Scope (b) Global scope  
(c) Module scope (d) Function Scope  
[Ans. (a) Local Scope]
- The process of subdividing a computer program into separate sub-programs is called  
(a) Procedural Programming  
(b) Modular programming  
(c) Event Driven Programming  
(d) Object oriented Programming  
[Ans. (b) Modular programming]
- Which of the following security technique that regulates who can use resources in a computing environment?  
(a) Password (b) Authentication  
(c) Access control (d) Certification  
[Ans. (c) Access control]
- Which of the following members of a class can be handled only from within the class?  
[Mar.-2020]  
(a) Public members  
(b) Protected members  
(c) Secured members  
(d) Private members  
[Ans. (d) Private members]

[18]

9. Which members are accessible from outside the class?

- (a) Public members
- (b) Protected members
- (c) Secured members
- (d) Private members

[Ans. (a) Public members]

10. The members that are accessible from within the class and are also available to its sub-classes is called [PTA-6]

- (a) Public members
- (b) Protected members
- (c) Secured members
- (d) Private members

[Ans. (b) Protected members]

### PART - II

ANSWER THE FOLLOWING QUESTIONS (2 MARKS)

1. What is a scope?

Ans. 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.

Ans. Essentially, variables are addresses (references, or pointers), to an object in memory. When you assign a variable with := to an instance (object), you're binding (or mapping) the variable to that instance. Multiple variable can be mapped to the same instance.

3. What is Mapping? [PTA-5]

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

4. What do you mean by Namespaces? [Govt. MQP-2019; PTA-4; Mar.-2020]

Ans. Namespaces are containers for mapping names of variables to objects.

Example : a := 5

Here the variable 'a' is mapped to the value '5'.

5. How Python represents the private and protected Access specifiers?

Ans. Python prescribes a convention of prefixing the name of the variable/method with single or double underscore to emulate the behaviour of protected and private access specifiers.

### PART - III

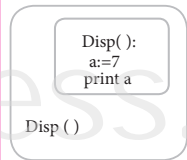
ANSWER THE FOLLOWING QUESTIONS (3 MARKS)

1. Define Local scope with an example.

Ans. (i) Local scope refers to variables defined in current function. Always, a function will first look up for a variable name in its local scope.

(ii) Only if it does not find it there, the outer scopes are checked.

(iii) Look at this example :

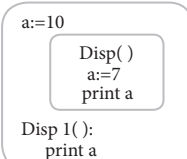
1. Disp(): 2. a:=7 3. print a 4. Disp()	Entire program 	Output of the Program 7
--	--	----------------------------

(iv) On execution of the above code the variable a displays the value 7, because it is defined and available in the local scope.

2. Define Global scope with an example. [PTA-6]

Ans. (i) A variable which is declared outside of all the functions in a program is known as Global variable.

(ii) This means, global variable can be accessed inside or outside of all the functions in a program. Consider the following example

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

(iii) On execution of the above code the variable a which is defined inside the function Disp() and then it displays 10, because a is defined in global scope.

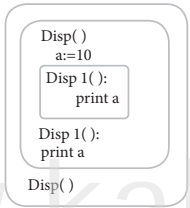
**3. Define Enclosed scope with an example.**

[PTA-3]

**Ans. (i)** All programming languages permit functions to be nested. A function (method) within another function is called nested function.

**(ii)** 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.

**(iii)** When a compiler or interpreter search for a variable in a program, it first search Local, and then search Enclosing scopes. Consider the following example

1. Disp(): 2. a:=10 3. Disp1(): 4. print a 5. Disp1() 6. print a 7. Disp()	Entire program 	Output of the Program 10 10
--	--	-----------------------------------

**4. Why access control is required?**

[PTA-1; HY-2019]

**Ans. (i)** Access control is a security technique that regulates who or what can view or use resources in a computing environment.

**(ii)** It is a fundamental concept in security that minimizes risk to the object.

**(iii)** In other words access control is a selective restriction of access to data.

**(iv)** In oops Access control is implemented through access modifiers.

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

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

**Ans. Output :**

Red Blue Green  
 Red Blue  
 Red

**Scope of Variables :**

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

**PART - IV**

**ANSWER THE FOLLOWING QUESTIONS**

**(5 MARKS)**

**1. Explain the types of scopes for variable or LEGB rule with example. [PTA-1; Sep-2020]**

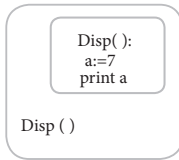
**Ans. Types of Variable Scope :**

There are 4 types of Variable Scope, let's discuss them one by one:

**Local Scope :**

**(i)** Local scope refers to variables defined in current function. Always, a function will first look up for a variable name in its local scope. Only if it does not find it there, the outer scopes are checked.

Look at this example

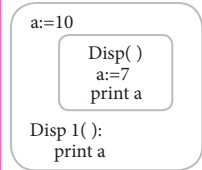
1. Disp(): 2. a:=7 3. print a 4. Disp()	Entire program 	Output of the Program 7
--	---	----------------------------

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

**Global Scope:**

**(i)** A variable which is declared outside of all the functions in a program is known as global variable.

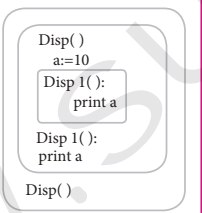
**(ii)** This means, global variable can be accessed inside or outside of all the functions in a program. Consider the following example

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

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

**Enclosed Scope :**

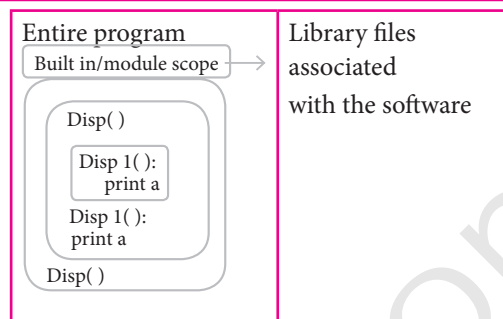
- (i) All programming languages permit functions to be nested. A function (method) with in another function is called nested function.
- (ii) A variable which is declared inside a function which contains another function definition with in it, the inner function can also access the variable of the outer function. This scope is called enclosed scope.
- (iii) When a compiler or interpreter search for a variable in a program, it first search Local, and then search Enclosing scopes. Consider the following example

1. Disp(): 2. a:=10 3. Disp1(): 4. print a 5. Disp1() 6. print a 7. Disp()	Entire program 	Output of the Program 10 10
--	---	-----------------------------------

- (iv) In the above example Disp1() is defined with in Disp(). The variable 'a' defined in Disp() can be even used by Disp1() because it is also a member of Disp().

**Built-in Scope :**

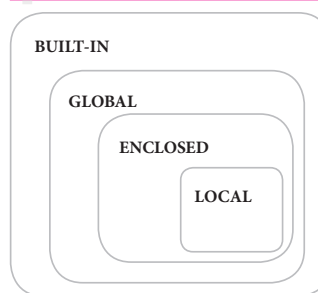
- (i) The built-in scope has all the names that are pre-loaded into the program scope when we start the compiler or interpreter.
- (ii) Any variable or module which is defined in the library functions of a programming language has Built-in or module scope. Consider the following example.



**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)



**2. Write any Five Characteristics of Modules.**  
[PTA-4, 6; HY-2019; Sep-2020]

**Ans.** The following are the desirable characteristics of a module.

- (i) Modules contain instructions, processing logic, and data.
- (ii) Modules can be separately compiled and stored in a library.
- (iii) Modules can be included in a program.
- (iv) Module segments can be used by invoking a name and some parameters.
- (v) Module segments can be used by other modules.



**CHAPTER SNAPSHOT**

- 4.1 Introduction to Algorithmic strategies
  - 4.1.1. Characteristics of an Algorithm
  - 4.1.2. Writing an Algorithm
  - 4.1.3. Analysis of Algorithm
- 4.2 Complexity of an Algorithm
  - 4.2.1. Time Complexity
  - 4.2.2. Space Complexity
- 4.3 Efficiency of an algorithm
  - 4.3.1. Method for determining Efficiency
  - 4.3.2. Space-Time tradeoff
  - 4.3.3. Asymptotic Notations
  - 4.3.4. Best, Worst, and Average ease Efficiency
- 4.4 Algorithm for Searching Techniques
  - 4.4.1. Linear Search
  - 4.4.2. Binary Search
- 4.5 Sorting Techniques
  - 4.5.1. Bubble sort algorithm
  - 4.5.2. Selection sort
  - 4.5.3. Insertion sort
- 4.6 Dynamic programming
  - 4.6.1. Fibonacci Series – An example
  - 4.6.2. Fibonacci Iterative Algorithm with Dynamic programming approach



## EVALUATION

### PART - I

#### 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 Khwarizmi is called? [PTA-6]

- (a) Flowchart (b) Flow  
(c) Algorithm (d) Syntax

[Ans. (c) Algorithm]

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

[Ans. (d) Selection sort]

3. Two main measures for the efficiency of an algorithm are [Mar.-2020]

- (a) Processor and memory  
(b) Complexity and capacity  
(c) Time and space (d) Data and space

[Ans. (c) Time and space]

4. The complexity of linear search algorithm is

- (a)  $O(n)$  (b)  $O(\log n)$   
(c)  $O(n^2)$  (d)  $O(n \log n)$

[Ans. (a)  $O(n)$ ]

5. From the following sorting algorithms which has the lowest worst case complexity?

- (a) Bubble sort (b) Quick sort  
(c) Merge sort (d) Selection sort

[Ans. (c) Merge sort]

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

- (a) Insertion sort (b) Selection sort  
(c) Bubble sort (d) Merge sort

[Ans. (b) Selection sort]

7. Time complexity of bubble sort in best case is [PTA-1]

- (a)  $\theta(n)$  (b)  $\theta(n \log n)$   
(c)  $\theta(n^2)$  (d)  $\theta(n(\log n)^2)$

[Ans. (a)  $\theta(n)$ ]

8. The  $\Theta$  notation in asymptotic evaluation represents

- (a) Base case (b) Average case  
(c) Worst case (d) NULL case

[Ans. (b) Average case]

9. If a problem can be broken into subproblems which are reused several times, the problem possesses which property?

- (a) Overlapping subproblems  
(b) Optimal substructure  
(c) Memoization  
(d) Greedy

[Ans. (a) Overlapping subproblems]

10. In dynamic programming, the technique of storing the previously calculated values is called? [HY-2019]

- (a) Saving value property  
(b) Storing value property  
(c) Memoization  
(d) Mapping

[Ans. (c) Memoization]

### PART - II

#### ANSWER THE FOLLOWING QUESTIONS

(2 MARKS)

1. What is an Algorithm? [Mar.-2020]

Ans. 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.

Ans. (i) Pseudo code is an informal high level description of the operations principle of a computer program or other algorithm.

(ii) It uses the structural conventions of a normal programming language, but is intended for human reading rather than machine reading.

3. Who is an Algorithmist?

Ans. Algorithmist may refer to

(i) A person skilled in the technique of performing basic decimal arithmetic, known as algorithm.

(ii) A person skilled in the design of algorithms.

(iii) An Algorithmic artist.

4. What is Sorting?

Ans. Sorting is any process of arranging information or data in an ordered sequence either in ascending or descending order. Various sorting techniques in algorithms are Bubble sort, Quick sort, Heap sort, Selection sort, Insertion sort.

5. What is searching? Write its types.

[Govt. MQP-2019; HY-2019]

**Ans.** A searching algorithm is the step-by-step procedure used to locate specific data among a collection of data. There are two type of searching are

- (i) Linear Search
- (ii) Binary Search

### PART - III

#### ANSWER THE FOLLOWING QUESTIONS

(3 MARKS)

1. List the characteristics of an algorithm.

- Ans.**
- (i) Input
  - (ii) Output
  - (iii) Finiteness
  - (iv) Definiteness
  - (v) Effectiveness
  - (vi) Correctness
  - (vii) Simplicity
  - (viii) Unambiguous
  - (ix) Feasibility
  - (x) Portable
  - (xi) Independent

2. Discuss about Algorithmic complexity and its types. [PTA-1]

**Ans.** The complexity of an algorithm  $f(n)$  gives the running time and/or the storage space required by the algorithm in terms of  $n$  as the size of input data.

- (i) **Time Complexity :** The Time complexity of an algorithm is given by the number of steps taken by the algorithm to complete the process.
- (ii) **Space Complexity :** Space complexity of an algorithm is the amount of memory required to run to its completion.

3. What are the factors that influence time and space complexity?

- Ans.**
- (i) **Time Factor** -Time is measured by counting the number of key operations like comparisons in the sorting algorithm.
  - (ii) **Space Factor** - Space is measured by the maximum memory space required by the algorithm.

4. Write a note on Asymptotic notation.

[QY-2019; Mar.-2020]

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

- (i) **Big O :** Big O is often used to describe the worst-case of an algorithm.
- (ii) **Big  $\Omega$  :** Big Omega is the reverse Big O, if Bi O is used to describe the upper bound (worst - case) of a asymptotic function, Big Omega is used to describe the lower bound (best-case).
- (iii) **Big  $\Theta$  :** When an algorithm has a complexity with lower bound = upper bound, say that an algorithm has a complexity  $O(n \log n)$  and  $\Omega(n \log n)$ , it's actually has the complexity  $\Theta(n \log n)$ , which means the running time of that algorithm always falls in  $n \log n$  in the best-case and worst-case.

5. What do you understand by Dynamic programming? [Sep-2020]

- Ans.**
- (i) Dynamic programming is an algorithmic design method that can be used when the solution to a problem can be viewed as the result of a sequence of decisions.
  - (ii) Dynamic programming approach is similar to divide and conquer. The given problem is divided into smaller and yet smaller possible sub-problems.
  - (iii) Dynamic programming is used whenever problems can be divided into similar sub-problems. So that their results can be re-used to complete the process.
  - (iv) Dynamic programming approaches are used to find the solution in optimized way. For every inner sub problem, dynamic algorithm will try to check the results of the previously solved sub-problems. The solutions of overlapped sub-problems are combined in order to get the better solution.

## PART - IV

### ANSWER THE FOLLOWING QUESTIONS

(5 MARKS)

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

Ans. [PTA-5; HY-2019]

<b>Input</b>	Zero or more quantities to be supplied.
<b>Output</b>	At least one quantity is produced.
<b>Finiteness</b>	Algorithms must terminate after finite number of steps.
<b>Definiteness</b>	All operations should be well defined. For example operations involving division by zero or taking square root for negative number are unacceptable.
<b>Effectiveness</b>	Every instruction must be carried out effectively.
<b>Correctness</b>	The algorithms should be error free.
<b>Simplicity</b>	Easy to implement.
<b>Unambiguous</b>	Algorithm should be clear and unambiguous. Each of its steps and their inputs/outputs should be clear and must lead to only one meaning.
<b>Feasibility</b>	Should be feasible with the available resources.
<b>Portable</b>	An algorithm should be generic, independent of any programming language or an operating system able to handle all range of inputs.
<b>Independent</b>	An algorithm should have step-by-step directions, which should be independent of any programming code.

#### 2. Discuss about Linear search algorithm.

[PTA-1; Mar.-2020]

Ans. (i) Linear search also called sequential search is a sequential method for finding a particular value in a list.

(ii) 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 :

- (i) Traverse the array using for loop
- (ii) In every iteration, compare the target search key value with the current value of the list.
  - If the values match, display the current index and value of the array
  - If the values do not match, move on to the next array element.
- (iii) If no match is found, display the search element not found.

#### Example

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

<b>index</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
<b>values</b>	<b>10</b>	<b>12</b>	<b>20</b>	<b>25</b>	<b>30</b>

#### Example 1 :

Input: values[] = {5, 34, 65, 12, 77, 35}  
target = 77

Output: 4

#### Example 2:

Input: values[] = {101, 392, 1, 54, 32, 22, 90, 93}  
target = 200

Output: -1 (not found)

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

Ans. **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.

#### Pseudo code for Binary search :

##### Start with the middle element:

- (i) If the search element is equal to the middle element of the array i.e., the middle value = number of elements in array/2, then return the index of the middle element.
- (ii) If not, then compare the middle element with the search value,
- (iii) If the search element is greater than the number in the middle index, then select the elements to the right side of the middle index, and go to Step-1.

# UNIT-II CORE PYTHON

## CHAPTER 5

## PYTHON-VARIABLES AND OPERATORS

### CHAPTER SNAPSHOT

5.1	Introduction	5.7	Tokens
5.2	Key features of Python	5.7.1.	Identifiers
5.3	Programming in Python	5.7.2.	Keywords
5.3.1	Interactive mode Programming	5.7.3	Operators
5.3.2	Script mode Programming	5.7.4	Delimiters
5.4	Input and Output Functions	5.7.5	Literals
5.4.1	The print() function	5.8	Python Data types
5.4.2	input() function	5.8.1	Number Data type
5.5	Comments in Python	5.8.2	Boolean Data type
5.6	Indentation	5.8.3	String Data type

### EVALUATION

#### PART - I

#### CHOOSE THE BEST ANSWER (1 MARK)

1. Who developed Python?

- (a) Ritche
- (b) Guido Van Rossum
- (c) Bill Gates
- (d) Sunder Pitchai

[Ans. (b) Guido Van Rossum]

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

- (a) >>>
- (b) <<<
- (c) #
- (d) <<

[Ans. (a) >>>]

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

[Ans. (d) Ctrl + N]

4. Which of the following character is used to give comments in Python Program? [PTA-6]

- (a) #
- (b) &
- (c) @
- (d) \$

[Ans. (a) #]

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

- (a) Semicolon (;)
- (b) Dollor (\$)
- (c) Comma (,)
- (d) Colon (:)

[Ans. (c) Comma (,)]

6. Which of the following is not a token?

- (a) Interpreter
- (b) Identifiers
- (c) Keyword
- (d) Operators

[Ans. (a) Interpreter]

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

- (a) break
- (b) while
- (c) continue
- (d) operators

[Ans. (d) operators]

8. Which operator is also called as Comparative operator? [PTA-1]

- (a) Arithmetic
- (b) Relational
- (c) Logical
- (d) Assignment

[Ans. (b) Relational]



9. Which of the following is not Logical operator?

- (a) and (b) or  
(c) not (d) Assignment

[Ans. (d) Assignment]

10. Which operator is also called as Conditional operator?

- (a) Ternary (b) Relational  
(c) Logical (d) Assignment

[Ans. (a) Ternary]

### PART - II

ANSWER THE FOLLOWING QUESTIONS

(2 MARKS)

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

Ans. (i) In Python, programs can be written in two namely Interactive mode and Script mode.

(ii) **Interactive mode** allows us to write codes in Python command prompt (>>>).

(iii) **Script mode** is used to create and edit python source file with the extension .py

2. Write short notes on Tokens. [PTA-4; HY-2019]

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

- (i) Identifiers,  
(ii) Keywords,  
(iii) Operators,  
(iv) Delimiters and  
(v) Literals.

3. What are the different operators that can be used in Python? [PTA-5, 6]

Ans. (i) Operators are special symbols which represent computations, conditional matching in programming.

(ii) The operators that can be used in Python.

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

Ans. Literal is a raw data given in a variable or constant. In Python, there are various types of literals.

- (i) **Numeric Literals** consists of digits and are immutable.  
(ii) **String literal** is a sequence of characters surrounded by quotes.  
(iii) **Boolean literal** can have any of the two values : True or False.

5. Write short notes on Exponent data.

Ans. An Exponent data contains decimal digit part, decimal point, exponent part followed by one or more digits.

Example: 12.E04, 24.e04.

### PART - III

ANSWER THE FOLLOWING QUESTIONS

(3 MARKS)

1. Write short notes on Arithmetic operator with examples.

Ans. (i) An arithmetic operator is a mathematical operator that takes two operands and performs a calculation on them. They are used for simple arithmetic.

(ii) Most computer languages contain a set of such operators that can be used within equations to perform different types of sequential calculations.

(iii) Python supports the following Arithmetic operators.

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

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

Ans. (i) In Python, = is a simple assignment operator to assign values to variable. Let **a = 5** and **b = 10** assigns the value 5 to **a** and 10 to **b** these two assignment statement can also be given as **a,b=5,10** that assigns the value 5 and 10 on the right to the variables **a** and **b** respectively.

(ii) There are various compound operators in Python like +=, -=, \*=, /=, %=, \*\*= and //= are also available.

Operator	Description	Example
Assume x=10		
=	Assigns right side operands to left variable	>>> x=10 >>> b="Computer"
+=	Added and assign back the result to left operand i.e. x=30	>>> x+=20 #x=x+20
-=	Subtracted and assign back the result to left operand i.e. x=25	>>> x-=5 # x=x-5
*=	Multiplied and assign back the result to left operand i.e. x=125	>>> x*=5 # x=x*5
/=	Divided and assign back the result to left operand i.e. x=62.5	>>> x/=2 # x=x/2
%=	Taken modulus (Remainder) using two operands and assign the result to left operand i.e. x=2.5	>>> x%=3 # x=x%3
**=	Performed exponential (power) calculation on operators and assign value to the left operand i.e. x=6.25	>>> X**=2 # x=x**2
//=	Performed floor division on operators and assign value to the left operand i.e. x=2.0	>>> x//=3

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

[PTA-1; Mar.-2020]

**Ans. (i)** Ternary operator is also known as conditional operator that evaluate something based on a condition being true or false.

(ii) 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]

**(iii) Example:**

min = 50 if 49 < 50 else 70 // min = 50

min = 50 if 49 > 50 else 70 // min = 70

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

**Ans. (i)** In Python strings, the backslash "\" is a special character, also called the "escape" character.

(ii) It is used in representing certain whitespace characters: "\t" is a tab, "\n" is a newline, and "\r" is a carriage return.

(iii) For example to print the message "It's raining", the Python command is

```
>>> print ("It's raining")
```

It's raining

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.**

- Ans. (i)** In Python, a string literal is a sequence of characters surrounded by quotes. Python supports single, double and triple quotes for a string.
- (ii)** 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.

**To test String Literals :**

# Demo Program to test String Literals

```
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)
```

# End of the Program

**Output:**

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

**PART - IV**

**ANSWER THE FOLLOWING QUESTIONS**

**(5 MARKS)**

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

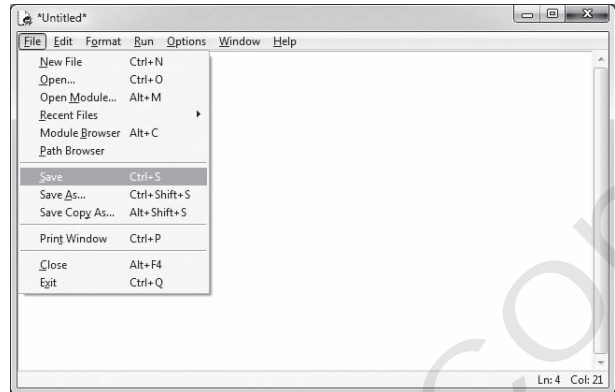
**Ans.** A script is a text file containing the Python statements. Python Scripts are reusable code. Once the script is created, it can be executed again and again without retyping. The Scripts are editable.

**Creating Scripts in Python :**

- (i)** Choose File → New File or press Ctrl + N in Python shell window.
- (ii)** An untitled blank script text editor will be displayed on screen.
- (iii)** Type the code in Script editor

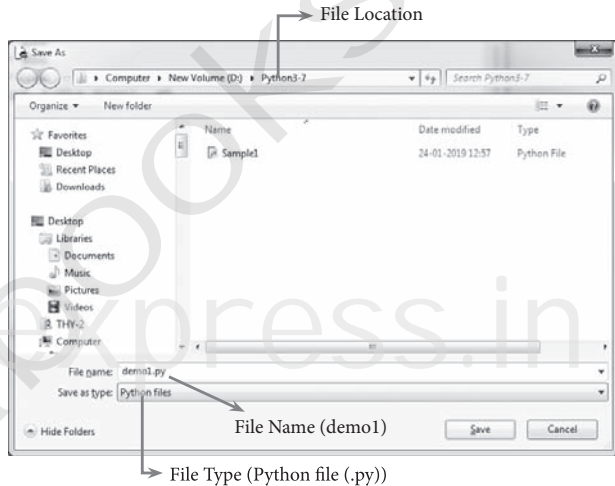
**Saving Python Script :**

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



To Save the file First time

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

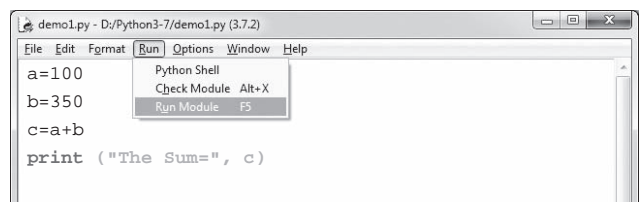


Save As Dialog Box

- (iii)** In the **Save As** dialog box, select the location where you want to save your Python code, and type the file name in **File Name** box. Python files are by default saved with extension **.py**. Thus, while creating Python scripts using Python Script editor, no need to specify the file extension.
- (iv)** Finally, click **Save** button to save your Python script.

**Executing Python Script :**

- (i)** Choose **Run** → **Run Module** or Press **F5**



To Execute Python Script

# CONTROL STRUCTURES

## CHAPTER SNAPSHOT

6.1 Introduction  
6.2 Control Structures  
6.2.1 Sequential Statement

6.2.2 Alternative or Branching Statement  
6.2.3. Iteration or Looping constructs  
6.2.4 Jump Statements in Python

## EVALUATION

### PART - I

#### CHOOSE THE BEST ANSWER (1 MARK)

1. How many important control structures are there in Python?  
(a) 3 (b) 4 (c) 5 (d) 6

[Ans. (a) 3]

2. elif can be considered to be abbreviation of  
(a) nested if (b) if..else  
(c) else if (d) if..elif

[Ans. (c) else if]

3. What plays a vital role in Python programming?

- (a) Statements (b) Control  
(c) Structure (d) Indentation

[Ans. (d) Indentation]

4. Which statement is generally used as a placeholder?

- (a) continue (b) break  
(c) pass (d) goto

[Ans. (c) pass]

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

[Ans. (c) Relational or Logical expression]

6. Which is the most comfortable loop?

- (a) do..while (b) while  
(c) for (d) if..elif

[Ans. (c) for]

7. What is the output of the following snippet?

```
i=1
while True:
    if i%3 ==0:
        break
    print(i,end=" ")
    i +=1
```

- (a) 12 (b) 123 (c) 1234 (d) 124

[Ans. (a) 12]

8. What is the output of the following snippet?

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

- (a) False (b) True  
(c) 0 (d) 1

[Ans. (b) True]

9. Which amongst this is not a jump statement ?

- (a) for (b) pass  
(c) continue (d) break

[Ans. (a) for]

10. Which punctuation should be used in the blank?

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

- statements-block 2  
(a) ; (b) : (c) :: (d) !

[Ans. (b) :]

### PART - II

#### ANSWER THE FOLLOWING QUESTIONS

(2 MARKS)

1. List the control structures in Python. [PTA-6]

Ans. There are three important control structures are,

- (i) Sequential  
(ii) Alternative or Branching  
(iii) Iterative or Looping

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

**Ans.** 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**

**Ans. Syntax :**

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

**4. Define control structure. [PTA-2]**

**Ans.** 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. [PTA-2; March-2020]**

**Ans.** range() generates a list of values starting from **start** till **stop-1**.

```
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.

**PART - III**

**ANSWER THE FOLLOWING QUESTIONS (3 MARKS)**

**1. Write a program to display [PTA-5]**

- A
- A B
- A B C
- A B C D
- A B C D E

**Ans.** for i in range (1, 6):  
    for j in range (65, 65 + i)  
        a=chr(j)  
    print

**2. Write note on if..else structure.**

**Ans. (i)** The **if..else** statement provides control to check the true block as well as the false block.

**(ii) if..else** statement thus provides two possibilities and the condition determines which BLOCK is to be executed.

**(iii) 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.**

**Ans. 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. [PTA-4; QY-2019]**

**Ans. Syntax:**

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

**5. List the differences between break and continue statements. [HY-2019]**

Ans.	Break	Continue
	The break statement terminates the loop containing it.	The continue statement is used to skip the remaining part of a loop.
	Control of the program flows to the statement immediately after the body of the loop.	Control of the program flows start with next iteration.
	<b>Syntax :</b> break	<b>Syntax :</b> continue

# PYTHON FUNCTIONS

## CHAPTER SNAPSHOT

- 7.01 Introduction
  - 7.1.1 Types of Functions
- 7.02 Defining Functions
  - 7.2.1 Syntax for User defined function
  - 7.2.2 Advantages of User-defined Functions
- 7.03 Calling a Function
- 7.04 Passing Parameters in Functions
- 7.05 Function Arguments
  - 7.5.1 Required Arguments
  - 7.5.2 Keyword Arguments
  - 7.5.3 Default Arguments
  - 7.5.4 Variable-Length Arguments
- 7.06 Anonymous Functions
  - 7.6.1 Syntax of Anonymous Functions
- 7.07 The return Statement
  - 7.7.1 Syntax of return
- 7.08 Scope of Variables
  - 7.8.1 Local Scope
  - 7.8.2 Global Scope
  - 7.8.3 Global and local variables
- 7.09 Functions using libraries
  - 7.9.1 Built-in and Mathematical functions
  - 7.9.2 Composition in functions
- 7.10 Python recursive functions

## EVALUATION

### PART - I

#### 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

[Ans. (c) Function]

2. A Function which calls itself is called as

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

[Ans. (b) Recursion]

3. Which function is called anonymous un-named function

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

[Ans. (a) Lambda]

4. Which of the following keyword is used to begin the function block? [Sep-2020]

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

[Ans. (d) def]

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

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

[Ans. (b) return]

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

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

[Ans. (c) : (colon)]

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

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

[Ans. (a) Required]

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

[Ans. (a) I is correct and II is 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)  $x\%4=0$

[Ans. (b)  $x\%4==0$ ]

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

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

[Ans. (c) def]

### PART - II

#### ANSWER THE FOLLOWING QUESTIONS

(2 MARKS)

1. What is function?

Ans. (i) Functions are named blocks of code that are designed to do one specific job.

(ii) Types of Functions are User defined, Built-in, lambda and recursion.

(iii) Function blocks begin with the keyword "def" followed by function name and parenthesis ().

2. Write the different types of function. [PTA-3]

Ans. (i) User - defined functions.

(ii) Built in functions.

(iii) Lambda functions.

(iv) Recursion functions.

3. What are the main advantages of function?

[HY-2019]

Ans. Main advantages of functions are

(i) It avoids repetition and makes high degree of code reusing.

(ii) It provides better modularity for your application.

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

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

Ans. 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.



6. What is base condition in recursive function? [PTA-6]

- Ans.** (i) A recursive function calls itself.  
(ii) The condition that is applied in any recursive function is known as base condition.  
(iii) A base condition is must in every recursive function otherwise it will continue to execute like an infinite loop.

7. How to set the limit for recursive function? Give an example. [PTA-5]

- Ans.** (i) Python stops calling recursive function after 1000 calls by default.  
(ii) So, it also allows you to change the limit using sys.setrecursionlimit (limit\_value).

**Example:**

```
import sys
sys.setrecursionlimit(3000)
def fact(n):
    if n == 0:
        return 1
    else:
        return n * fact(n-1)
print(fact(2000))
```

### PART - III

#### ANSWER THE FOLLOWING QUESTIONS

(3 MARKS)

1. Write the rules of local variable.

**Ans. Rules of local variable :**

- (i) A variable with local scope can be accessed only within the function/block that it is created in.  
(ii) When a variable is created inside the function/block, the variable becomes local to it.  
(iii) A local variable only exists while the function is executing.  
(iv) The formate arguments are also local to function.

2. Write the basic rules for global keyword in python. [PTA-4]

**Ans. Rules of global Keyword :** The basic rules for global keyword in Python are:

- (i) When we define a variable outside a function, it's global by default. You don't have to use global keyword.  
(ii) We use global keyword to read and write a global variable inside a function.  
(iii) Use of global keyword outside a function has no effect.

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

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

**Example:**

```
x = 0 # global variable
def add():
    global x
    x = x + 5 # increment by 2
    print ("Inside add() function x value is :", x)
add()
print ("In main x value is :", x)
```

**Output :**

```
Inside add() function x value is : 5
In main x value is : 5 # value of x changed
                        outside the function
```

4. Differentiate ceil() and floor() function.

**Ans.** [PTA-2]

S.No.	ceil ()	floor ()
(i)	Returns the smallest integer greater than or equal to x.	Returns the largest integer less than or equal to x.
(ii)	Syntax : math.ceil(x)	Syntax : math.floor(x)

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

**Ans. Code :**

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

**Output :**

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

6. What is composition in functions?

**Ans. (i)** The value returned by a function may be used as an argument for another function in a nested manner.

(ii) This is called **composition**. For example, if we wish to take a numeric value or an expression as a input from the user, we take the input string from the user using the function **input()** and apply **eval()** function to evaluate its value

**7. How recursive function works? [Mar.-2020]**

**Ans. (i)** Recursive function is called by some external code.

**(ii)** If the base condition is met then the program gives meaningful output and exits.

**(iii)** Otherwise, function does some required processing and then calls itself to continue recursion.

**8. What are the points to be noted while defining a function? [Govt. MQP-2019]**

**Ans.** When defining functions there are multiple things that need to be noted;

**(i)** Function blocks begin with the keyword "def" followed by function name and parenthesis ().

**(ii)** Any input parameters or arguments should be placed within these parentheses when you define a function.

**(iii)** The code block always comes after colon(:) and is indented.

**(iv)** The statement "return [expression]" exits a function, optionally passing back an expression to the caller. A "return" with no arguments is the same as return None.

**PART - IV**

**ANSWER THE FOLLOWING QUESTIONS**

**(5 MARKS)**

**1. Explain the different types of function with an example. [Govt. MQP-2019; PTA-4]**

**Ans.** Functions are named blocks of code that are designed to do one specific job.

**Types of Functions :**

- (i) User Defined Function
- (ii) Built-in Function
- (iii) Lambda Function
- (iv) Recursion Function

**(i) 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 ().

**Example :**

```
def area(w,h):  
    return w * h  
print (area (3,5))
```

**(ii) Built-in Functions :** Functions which are using Python libraries are called Built-in function.

```
x=20  
y=-23.2  
print('x = ', abs(x))  
print('y = ', abs(y))
```

**Output:**

```
x = 20  
y = 23.2
```

**(iii) Lambda Functions :**

- 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.

**Example :**

```
sum = lambda arg1, arg2: arg1 + arg2  
print ("The Sum is ", sum(30,40))  
print ("The Sum is :", sum(-30,40))
```

**Output :**

```
The Sum is : 70  
The Sum is : 10
```

**(iv) Recursive function :** Functions that calls itself is known as recursive.

**Overview of how recursive function works :**

- (i) Recursive function is called by some external code.
- (ii) If the base condition is met then the program gives meaningful output and exits.
- (iii) Otherwise, function does some required processing and then calls itself to continue recursion.

**Example :**

```
def fact(n):  
    if n == 0:  
        return 1  
  
    else:  
        return n * fact (n-1)  
  
print (fact (0))  
print (fact (5))
```

**Output:**

```
1  
120
```

2. Explain the scope of variables with an example. [PTA-3; HY-2019]

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

**Local Scope :** A variable declared inside the function's body or in the local scope is known as local variable.

**Rules of local variable :**

- (i) A variable with local scope can be accessed only within the function/block that it is created in.
- (ii) When a variable is created inside the function/block, the variable becomes local to it.
- (iii) A local variable only exists while the function is executing.
- (iv) The formal arguments are also local to function.
- (v) **Example:** Create a Local Variable

```
def loc():
    y=0 # local scope
    print(y)
loc()
```

**Output:**

0

**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 :**

The basic rules for global keyword in Python are:

- (i) When we define a variable outside a function, it's global by default. You don't have to use global keyword.
- (ii) We use global keyword to read and write a global variable inside a function.
- (iii) Use of global keyword outside a function has no effect.

**Example :** Accessing global Variable From Inside a Function

```
c = 1 # global variable
def add():
    print(c)
add()
```

**Output:**

1

3. Explain the following built-in functions.

- (a) id()                      (b) chr()
- (c) round()                (d) type()
- (e) pow()

**Ans.**

(a) [PTA-4, 6; QY-2019; Mar.-2020]

Function	Description	Syntax	Example
id ()	id ( ) Return the "identity" of an object. i.e. the address of the object in memory.	id (object)	x=15 y='a' print ('address of x is:',id (x)) print ('address of y is:',id (y))
	<b>Note:</b> The address of x and y may differ in your system.		<b>Output:</b> address of x is : 1357486752 address of y is : 13480736

(b) [PTA-4]

Function	Description	Syntax	Example
chr ( )	Returns the Unicode character for the given ASCII value. This function is inverse of ord() function.	chr (i)	c=65 d=43 print (chr (c)) print (chr (d))
			<b>Output:</b> A +

(c)

Function	Description	Syntax	Example
round ( )	Returns the nearest integer to its input. <b>1.</b> First argument (number) is used to specify the value to be rounded.	round (number [,ndigits])	x= 17.9 y= 22.2 z= -18.3 print ('x value is rounded to', round (x)) print ('y value is rounded to', round (y)) print ('z value is rounded to', round (z))

(d) [PTA-4; QY-2019]

Function	Description	Syntax	Example
type ( )	Returns the type of object for the given single object. <b>Note:</b> This function used with single object parameter.	type (object)	x= 15.2 y= 'a' s= True print (type (x)) print (type (y)) print (type (s)) <b>Output:</b> <class 'float'> <class 'str'> <class 'bool'>

(e)

Function	Description	Syntax	Example
pow ( )	Returns the computation of ab i.e. (a**b) a raised to the power of b.	pow (a,b)	a= 5 b= 2 c= 3.0 print (pow (a,b)) print (pow (a,c)) print (pow (a+b,3)) <b>Output:</b> 25 125.0 343

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

**Ans. Program :**

```
def lcm(x, y):
    if x>y:
        greater = x
    else:
        greater = y
    while (True):
        if ((greater % x == 0) and (greater % y == 0)):
            lcm = greater
            break
        greater += 1
    return lcm
a = int (input ("Enter first number :"))
b = int (input ("Enter second number :"))
print ("The LCM of ", a, "and", b, "is", LCM(a,b))
```

5. Explain recursive function with an example.

[PTA-5]

**Ans. (i)** A Functions that calls itself is known as recursive.

**(ii)** When a function calls itself is known as recursion.

**(iii)** Recursion works like loop but sometimes it makes more sense to use recursion than loop.

**(iv)** Imagine a process would iterate indefinitely if not stopped by some condition is known as infinite iteration.

**(v)** The condition that is applied in any recursive function is known as base condition.

**(vi)** A base condition is must in every recursive function otherwise it will continue to execute like an infinite loop.

**(vii)** Python stops calling recursive function after 1000 calls by default.

**(viii)** So, It also allows you to change the limit using sys.setrecursionlimit (limit\_value).

**Overview of how recursive function works :**

**(i)** Recursive function is called by some external code.

**(ii)** If the base condition is met then the program gives meaningful output and exits.

**(iii)** Otherwise, function does some required processing and then calls itself to continue recursion.

Here is an example of recursive function used to calculate factorial.

# CHAPTER 8

## STRINGS AND STRING MANIPULATION

### CHAPTER SNAPSHOT

8.1	Introduction	8.6	String Formatting Operators
8.2	Creating Strings	8.7	Formatting characters
8.3	Accessing characters in a String	8.8	The format() function
8.4	Modifying and Deleting Strings	8.9	Built-in String functions
8.5	String Operators	8.10	Membership Operators

### EVALUATION

#### PART - I

#### 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

[Ans. (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

[Ans. (c) Type error]

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

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

[Ans. (a) +]

4. Defining strings within triple quotes allows creating:

- (a) Single line Strings (b) Multiline Strings  
(c) Double line Strings (d) Multiple Strings

[Ans. (b) Multiline Strings]

5. Strings in python:

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

[Ans. (c) Immutable]

6. Which of the following is the slicing operator? [PTA-1]

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

[Ans. (b) []]

7. What is stride? [PTA-2]

- (a) index value of slide operation  
(b) first argument of slice operation  
(c) second argument of slice operation  
(d) third argument of slice operation

[Ans. (d) third argument of slice operation]

8. Which of the following formatting character is used to print exponential notation in upper case? [PTA-5]

- (a) %e (b) %E (c) %g (d) %n

[Ans. (b) %E]

9. Which of the following is used as placeholders or replacement fields which get replaced along with format() function? [PTA-4]

- (a) {} (b) <> (c) ++ (d) ^^

[Ans. (a) {}]

10. The subscript of a string may be:

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

[Ans. (d) Either (a) or (b)]



**PART - II**

**ANSWER THE FOLLOWING QUESTIONS**

**(2 MARKS)**

**1. What is String?**

**Ans. (i)** String is a data type in python, which is used to handle array of characters.

**(ii)** String is a sequence of Unicode characters that may be a combination of letters, numbers, or special symbols enclosed within single, double or even triple quotes.

**(iii) Example :**

'Welcome to learning Python'  
"Welcome to learning Python"  
" "Welcome to learning Python" "

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

**Ans. (i)** Yes we can modify the string by the following method.

**(ii)** A new string value can be assign to the existing string variable.

**(iii)** When defining a new string value to the existing string variable.

**(iv)** Python completely overwrite new string on the existing string.

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

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

```
>>> str1="How about you"
>>> print (str1)
How about you
>>> del str1
>>> print (str1)
Traceback (most recent call last):
  File "<pyshell#14>", line 1, in <module>
    print (str1)
```

**NameError: name 'str1' is not defined**

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

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

**Ans. Output :** School School School

**5. What is slicing?** [PTA-6]

**Ans. (i)** Slice is a substring of a main string. A substring can be taken from the original string by using [] operator and index or subscript values.

**(ii)** Thus, [] is also known as slicing operator. Using slice operator, you have to slice one or more substrings from a main string.

**General format of slice operation :**

Str[start : end]

**PART - III**

**ANSWER THE FOLLOWING QUESTIONS**

**(3 MARKS)**

**1. Write a Python program to display the given pattern.** [Govt. MQP-2019]

```
COMPUTER
COMPUTE
COMPUT
COMPU
COMP
COM
CO
C
```

**Ans.** str1 = "COMPUTER"

```
index = len(str1)
for i in str1:
    print (str1[: index])
    index - = 1
```

**2. Write a short about the followings with suitable example:**

**(a) capitalize() (b) swapcase()**

[PTA-1, 3; Sep-2020]

**Ans.**

Syntax	Description	Example
(a) capitalize()	Used to capitalize the first character of the string	>>> city="chennai" >>> print(city.capitalize()) <b>Chennai</b>
(b) swapcase()	It will change case of every character to its opposite case vice-versa.	>>> str1="tAmiL NaDu" >>> print(str1.swapcase()) <b>TaMIl nAdU</b>

PART - IV

ANSWER THE FOLLOWING QUESTIONS

(5 MARKS)

1. Explain about string operators in python with suitable example. [PTA-2; HY-2019]

Ans. **String Operators** : Python provides the following operators for string operations. These operators are useful to manipulate string.

(i) **Concatenation (+)** : Joining of two or more strings is called as Concatenation. The plus (+) operator is used to concatenate strings in python.

**Example :**

```
>>> "welcome" + "Python"
'welcomePython'
```

(ii) **Append (+ =)** : Adding more strings at the end of an existing string is known as append. The operator += is used to append a new string with an existing string.

**Example :**

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

*Welcome to Learn Python*

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

**Example :**

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

(iv) **String slicing :**

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

**General format of slice operation :**

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

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

**Example :**

**slice a single character from a string**

```
>>>str1="THIRUKKURAL"
>>>print (str1[0])
```

**Output :**

T

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

```
str1 = "welcome"
str2 = "to school"
str3 = str1[:2]+str2[len(str2)-2:]
print(str3)
```

Ans. weol

4. What is the use of format()? Give an example. [HY-2019]

Ans. (i) The format() function used with strings is very versatile and powerful function used for formatting strings.

(ii) The curly braces {} are used as placeholders or replacement fields which get replaced along with format() function.

(iii) **Example :**

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

**Output :**

Number 1: 34  
Number 2: 54  
The sum of 34 and 54 is 88

5. Write a note about count() function in python. Ans.

Syntax	Description	Example
count (str, beg, end)	Returns the number of substrings occurs within the given range. Remember that substring may be a single character. Range (beg and end) arguments are optional. If it is not given, python searched in whole string. Search is case sensitive.	<pre>&gt;&gt;&gt; str1="Raja Raja Chozhan" &gt;&gt;&gt; print(str1.count('Raja')) 2 &gt;&gt;&gt; print(str1.count('r')) 0 &gt;&gt;&gt; print(str1.count('R')) 2 &gt;&gt;&gt; print(str1.count('a')) 5 &gt;&gt;&gt; print(str1.count('a',0,5)) 2 &gt;&gt;&gt; print(str1.count('a',11)) 1</pre>

(v) **Stride when slicing string :**

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

**Example :**

```
>>>str1="Welcome to learn Python"
>>>print (str1[10:16])
>>>print (str1[::-2])
```

**Output :**

```
Learn
nhy re teolW
```

**HANDS ON EXPERIENCE**

1. Write a python program to find the length of a string.

**Ans.** str = input ("Enter a string")  
print (len(str))

**Output :**

```
Enter a string : HELLO
5
```

2. Write a program to count the occurrences of each word in a given string.

**Ans.** def word\_conunt(str):  
counts=dict()  
words=str.split()  
for word in words:  
if word in counts:  
counts[word]+=1  
else:  
counts[word]=1  
return counts  
print(word\_count('the quick brown fox jumps over the lazy dog.'))

**Output :**

```
{'th' :2, 'jumps' : 1, 'brown' : 1, 'lazy' : 1, 'fox' : 1, 'over' : 1, 'quick' : 1, 'dog' : 1}
```

3. Write a program to add a prefix text to all the lines in a string.

**Ans.** import textwrap  
text =  
"""Strings are immutable. Slice is a substring of a main string. Stride is a third argument in slicing operation"""  
text\_without\_Indentation = textwrap.  
dedent(text)

```
wrapped=textwrap.fill(text_without_Indentation, width=50)
print(textwrap.indent(wrapped, '*')
print ( )
```

**Output :**

\* Strings are immutable. Slice is a substring of a main string. Stride is a third argument in slicing operation

4. Write a program to print integers with '\*' on the right of specified width.

**Ans.** x = 1 2 3  
print("original number:", x)  
print("formatted number(right padding, width 6) : "+"{:\*<7d}".format(x));

**Output :**

```
original number : 1 2 3
formatted number(right padding, width 6):
1 2 3 ***
```

5. Write a program to create a mirror image of the given string. For example, "wel" = "lew".

**Ans.** str1= input ("Enter a string")  
str2 = ''  
index = -1  
for i in str1:  
str2 = str1[index]  
index - = 1  
print ("The given string = {} \n The Reversed string = {}".format(str1, str2))

**Output :**

```
Enter a string : welcome
The given string = welcome
The Reversed string = emoclew
```

6. Write a program to removes all the occurrences of a give character in a string.

**Ans.** def removechar(s,c):  
# find total no of occurrence of a character  
counts = s.count(c)  
# convert into list of characters  
s = list(s)  
# keep looping until counts become 0  
while counts :  
# remove char. from list  
s.remove(c)  
counts - = 1  
# join remaining characters  
s = ".join(s)  
print(s)  
s = "Python programming"  
remove char(s, 'p')

**Output :**

```
ython rogramming
```

7. Write a program to append a string to another string without using += operator.

**Ans.** s1 = input("Enter first string :")  
s2 = input("Enter second string :")  
print('concatenated strings =', " ".join([s1,s2]))

**Output :**

Enter the first string : Tamil  
Enter the second string : Nadu  
concatenated strings = Tamil Nadu

8. Write a program to swap two strings.

**Ans.** print("Enter 'x' for exit.")  
string1 = input("Enter first string :")  
if string1 == 'x':  
exit();  
else:  
string2=input("Enter second string :")  
print("\n Both strings before swap : ")  
print("First string =", string1)  
print("Second string = ", string2)  
temp = string1  
string1 = string2  
string2 = temp  
print("\n Both strings after swap :")  
print("First string =", string1)  
print("Second string =", string2)

**Output :**

Enter 'x' for exit  
Enter first string : code  
Enter second string : python  
Both strings before swap:  
First string = code  
Second string = python  
Both strings after swap:  
First string = python  
Second string = code

9. Write a program to replace a string with another string without using replace().

**Ans.** s1 = input("Enter the string to be replaced :")  
s2 = input("Enter string to replace with")  
s1 = s2  
print("Replaced strings is :, s1)

**Output :**

Enter the string to be replaced : Computer  
Enter the string to replace with : repcomputer  
Replaced string is repcomputer

10. Write a program to count the number of characters, words and lines in a given string.

**Ans.** string = input("Enter string :")  
char = 0  
word = 0  
line = 0  
for i in string :  
char = char + 1  
if(i == ' '):  
word = word + 1  
elif(i == '\n'):  
line = line + 1  
print("Number of words in the string :")  
print(word)  
print("Number of characters in the string :")  
print(char)  
print("Number of lines in the string :")  
print(line)

**Output :**

Enter string : weclome to learning python  
Number of words in the string : 4  
Number of characters in the string : 26  
Number of lines in the string : 1

### PTA QUESTIONS AND ANSWERS

#### 1 MARK

1. Which command can be used to remove entire string variable in Python? [PTA-3]

- (a) rem (b) remove  
(c) del (d) delete

[Ans. (c) del]

2. What will be the output of the following snippet? [PTA-6]

str1="COMPUTER"  
print(str1[::2])

- (a) ER (b) CO  
(c) OPTR (d) CMUE

[Ans. (c) OPTR]

#### 2 MARKS

1. What will be the output of the following Python code? [PTA-1]

Str1 = "Madurai"  
print(Str1\*3)

**Ans. Output :**

Madurai Madurai Madurai

**2.** Write the usage of the following format string characters.

- (i) % c
- (ii) % d (or) % i
- (iii) % s

**Ans.** (i) character  
 (ii) Signed decimal integer  
 (iii) String

**3.** Fill in the blanks

	Format character	Usage
i)	_____	Unsigned decimal integer
ii)	% 0	_____
iii)	_____	Hexadecimal integer
iv)	% i	_____

**Ans.** (i) % u  
 (ii) octal integer  
 (iii) % c or %x  
 (iv) signed decimal integer.

**4.** Write the formats string characters for the following

- (i) Exponential notation
- (ii) Floating point numbers
- (iii) Short numbers in exponential notation

**Ans.** (i) % e or % E  
 (ii) % f  
 (iii) % g or % G

**5.** Write the description for the following escape sequence.

- (i) \r
- (ii) \000
- (iii) \v

**Ans.** (i) carriage return  
 (ii) character with octal value  
 (iii) vertical tab

**6.** Write the output of the following statements.

- (i) `print (len ("Corporation"))`
- (ii) `print ("school", Capitalize ())`
- (iii) `print ("Welcome" center (15, '*'))`

**Ans.** (i) 11  
 (ii) School  
 (iii) \* \* \* \* Welcome \* \* \* \*

**7.** Write the output of the following statements.

- `str1 = 'mammals'`
- (i) `str1.find ('ma')`
  - (ii) `str1.find ('ma', 2)`
  - (iii) `str1.find ('ma', 2, 4)`
  - (iv) `str1.find ('ma', 2, 5)`

**Ans.** (i) 0  
 (ii) 3  
 (iii) - 1  
 (iv) 3

**8.** Write the output of the following statements.

- (i) `'Python 2. 3' . isalpha ()`
- (ii) `'Python Program' . isalnum ()`
- (iii) `'Python' . isupper ()`

**Ans.** (i) False  
 (ii) False  
 (iii) False

**9.** Write the output for the following statement.

- (i) `print ("PYTHON" . lower ())`
- (ii) `print ("PYTHON" . islower ())`
- (iii) `print ("PYTHON" . isupper ())`
- (iv) `print ("PYTHON" . upper ())`

**Ans.** (i) python  
 (ii) false  
 (iii) false  
 (iv) PYTHON



10. Write a note on

- (i) isalnum ()                      (ii) isalpha ()                      (iii) isdigit ()

Ans.

	Syntax	Description	Example
(i)	isalnum ( )	Returns True if the string contains only letters and digit. It returns False. If the string contains any special character like _, @, #, *, etc.	>>>str1='Save Earth' >>>str1.isalnum() <i>False</i> The function returns False as space is an alphanumeric character. >>>'Save1Earth'.isalnum() <i>True</i>
(ii)	isalpha ( )	Returns True if the string contains only letters. Otherwise return False.	>>>'Click123'.isalpha() False >>>'python'.isalpha ( ) <i>True</i>
(iii)	isdigit ( )	Returns True if the string contains only numbers. Otherwise it returns False.	>>> str1='Save Earth' >>>print(str1.isdigit ( )) <i>False</i>

11. Write the out put for the following statement.

str1 = "Raja Raja Chozhan"

- (i) print (str1 . count ('Raja'))  
(ii) print (str1 . count ('R'))  
(iii) print (str1 . count ('A'))  
(iv) print (str1 . count ('a'))  
(v) print (str1 . count ('a', 0, 5))  
(vi) print (str1 . count ('a', 11))

Ans. (i) 2                      (ii) 2  
(iii) 0                      (iv) 5  
(v) 2                      (vi) 1

LONG ANSWERS

5 MARKS

1. Write a python program to print the following pattern

```
*
**
***
****
*****
```

Ans. str1='\*'  
i=1  
while i<=5:  
    print (str1\*i)  
    i+=1

2. Write a python program to display the number of vowels and consonants in the given string.

Ans. str1=input ("Enter a string: ")  
str2="aAeEiIoOuU"  
v,c=0,0  
for i in str1:  
    if i in str2:  
        v+=1  
    else:  
        c+=1  
print ("The given string contains { } vowels and { } consonants".format(v,c))

3. Write a python program to create an Abecedarian series, (Abecedarian refers list of elements appear in alphabetical order).

Ans. str1="ABCDEFGH"  
str2="ate"  
for i in str1:  
    print ((i+str2),end='\t')  
**Output :**  
Aate Bate Cate Date Eate Fate Gate Hate

4. Write a python program that accept a string from the user and display the same after removing vowels from it.

**Ans.** def rem\_vowels(s):  
temp\_str=""  
for i in s:  
if i in "aAeEiIoOuU":  
pass  
else:  
temp\_str+=i  
print ("The string without vowels: ", temp\_str)  
str1= input ("Enter a String: ")  
rem\_vowels (str1)

5. Write a python program that searches and counts the occurrences of a character in string.

**Ans.** def count(s, c):  
c1=0  
for i in s:  
if i == c:  
c1+=1  
return c1  
str1=input ("Enter a String: ")

```
ch=input ("Enter a character to be searched: ")  
cnt=count (str1, ch)  
print ("The given character {} is occurs {} times  
in the given string".format(ch,cnt))
```

**Output:**

```
Enter a String: Software Engineering  
Enter a character to be searched: e  
The given character e is occurs 3 times in the  
given string
```

6. Write the program to display the following pattern

```
* * * * *  
* * * *  
* * *  
* *  
*  
*
```

**Ans.** str1 = '\*'  
i = 5  
while (i >= 1)  
print (str1 \* i)  
i = 1



CHAPTER  
**15**

# DATA MANIPULATION THROUGH SQL

## CHAPTER SNAPSHOT

- 15.1 Introduction
- 15.2 SQLite
- 15.3 Creating a Database using SQLite
  - 15.3.1 Creating a Table
  - 15.3.2 Adding Records
- 15.4 SQL Query Using Python
  - 15.4.1 SELECT Query
  - 15.4.2 CLAUSES IN SQL
- 15.5 The SQL AND, OR and NOT Operators
- 15.6 Querying A Date Column
- 15.7 Aggregate Functions
  - 15.7.1 COUNT ()function
  - 15.7.2 AVG ()
  - 15.7.3 SUM ()
  - 15.7.4 MAX () And MIN () Functions
- 15.8 Updating A Record
- 15.9 Deletion Operation
- 15.10 Data input by User
- 15.11 Using Multiple Table for Querying
- 15.12 Integrating Query With Csv File
- 15.13 Table List

## EVALUATION

### PART - I

#### CHOOSE THE BEST ANSWER (1 MARK)

- Which of the following is an organized collection of data? [Mar.-2020]  
(a) Database (b) DBMS  
(c) Information (d) Records  
[Ans. (a) Database]
- SQLite falls under which database system?  
(a) Flat file database system  
(b) Relational Database system  
(c) Hierarchical database system  
(d) Object oriented Database system  
[Ans. (b) Relational Database system]
- Which of the following is a control structure used to traverse and fetch the records of the database? [PTA-4]  
(a) Pointer (b) Key  
(c) Cursor (d) Insertion point  
[Ans. (c) Cursor]
- Any changes made in the values of the record should be saved by the command  
(a) Save (b) Save As  
(c) Commit (d) Oblige  
[Ans. (c) Commit]
- Which of the following executes the SQL command to perform some action?  
(a) execute() (b) key()  
(c) cursor() (d) run()  
[Ans. (a) execute()]
- 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()  
[Ans. (c) AVG()]
- The function that returns the largest value of the selected column is  
(a) MAX() (b) LARGE()  
(c) HIGH() (d) MAXIMUM()  
[Ans. (a) MAX()]

- Which of the following is called the master table? [PTA-6]  
(a) sqlite\_master  
(b) sql\_master  
(c) main\_master  
(d) master\_main [Ans. (a) sqlite\_master]
- The most commonly used statement in SQL is  
(a) cursor (b) select [HY-2019]  
(c) execute (d) commit  
[Ans. (b) select]
- Which of the following clause avoid the duplicate? [PTA-5]  
(a) Distinct (b) Remove  
(c) Where (d) GroupBy  
[Ans. (a) Distinct]

### PART - II

#### ANSWER THE FOLLOWING QUESTIONS (2 MARKS)

- Mention the users who use the Database.  
**Ans.** Users of database can be human users, other programs or applications.
- Which method is used to connect a database? Give an example.  
**Ans.** Create a connection using connect () method and pass the name of the database File.  
**Example :**

```
import sqlite3
# connecting to the database
connection = sqlite3.connect("Academy.db")
# cursor
cursor = connection.cursor()
```
- What is the advantage of declaring a column as "INTEGER PRIMARY KEY"? [Mar.-2020]  
**Ans.** If a column of a table is declared to be an INTEGER PRIMARY KEY, then whenever a NULL is used as an input for this column, the NULL will be automatically converted into an integer which will be one larger than the highest value so far used in that column.  
If the table is empty, the value 1 will be used.

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

**Ans.** 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?

**Ans.** The fetchall() method is used to fetch all rows from the database table.

**Example :** result = cursor.fetchall()

### PART - III

#### ANSWER THE FOLLOWING QUESTIONS

(3 MARKS)

1. What is SQLite? What is its advantage?

[PTA-1; HY-2019]

**Ans. (i)** SQLite is a simple relational database system, which saves its data in regular data files or even in the internal memory of the computer.

**(ii)** It is designed to be embedded in applications, instead of using a separate database server program such as MySQL or Oracle.

**(iii)** 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(). [PTA-4; Mar.-2020]

**Ans.**

	fetchone()	fetchmany()
<b>(i)</b>	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.
<b>(ii)</b>	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.

[Govt. MQP-2019]

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

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

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

**OUTPUT:**

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

4. Read the following details. Based on that write a python script to display department wise records [PTA-5, 6]

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

**Ans.**

```
import sqlite3
connection = sqlite3.connect("organization.db")
cursor = connection.cursor()
cursor.execute("SELECT * FROM Employee
GROUPBY 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 descending order of

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

**Ans.**

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

result = cursor.fetchall()
print(result)
```



## PART - IV

### ANSWER THE FOLLOWING QUESTIONS

(5 MARKS)

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

**Ans. (i)** SQLite is a simple relational database system, which saves its data in regular data files or even in the internal memory of the computer.

**(ii)** It is designed to be embedded in applications, instead of using a separate database server program such as MySQL or Oracle.

#### Advantages :

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

#### To use SQLite,

**Step 1 :** import sqlite3

**Step 2 :** create a connection using connect () method and pass the name of the database file

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

**(ii)** 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.

**(iii)** Cursor in step 3 is a control structure used to traverse and fetch the records of the database.

**(iv)** Cursor has a major role in working with Python. All the commands will be executed using cursor object only.

**(v)** To create a table in the database, create an object and write the SQL command in it.

**Example :** sql\_comm = "SQL statement"

**(vi)** For executing the command use the cursor method and pass the required sql command as a parameter. Many number of commands can be stored in the sql\_comm and can be executed one after other.

**(vii)** Any changes made in the values of the record should be saved by the command "Commit" before closing the "Table connection".

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

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

**Ans.** Assume database name, is shop.db and table name is electronics for the given table.

#### Python Script :

```
import sqlite3
connection = sqlite3.connect ("shop.db")
cursor = connection.cursor ()
cursor.execute ("SELECT * FROM
                                electronics")
result = cursor.fetchall ()
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. [PTA-5]

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

#### Example :

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT
                GENDER,COUNT(GENDER) FROM Student
                GROUP BY GENDER HAVING
                COUNT(GENDER)>3")
```

```
result = cursor.fetchall()
co = [i[0] for i in cursor.description]
print(co)
print(result)
```

#### Output :

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

12<sup>th</sup>  
STD

GOVT. SUPPLEMENTARY EXAM - SEP - 2020  
COMPUTER SCIENCE (with Answers)

Reg. No.

--	--	--	--	--	--	--	--

TIME ALLOWED : 3.00 Hours]

PART - III

[MAXIMUM MARKS : 70

**Instructions :**

- 1) Check the question paper for fairness of printing. If there is any lack of fairness, inform the Hall Supervisor immediately.
- 2) Use **Blue** or **Black** ink to write and underline and pencil to draw diagrams

**PART - I**

**Note :** (i) Answer all the questions.

- (ii) Choose the most appropriate answer from the given **four** alternatives and write the option code and the corresponding answer. **[15 × 1 = 15]**

1. Which is the basic building block of computer programs?  
(a) Argument (b) Parameter  
(c) Subroutine (d) Interface
2. Which functions build the abstract data type?  
(a) Constructors (b) Data  
(c) List (d) Tuple
3. The process of binding a variable name with an object is called :  
(a) Scope (b) Mapping  
(c) Namespaces (d) Memory
4. Which one of the following is not a factor to measure the execution time of an algorithm?  
(a) Speed of the machine  
(b) Operating system  
(c) Programming language used  
(d) Selection
5. Which key is pressed to execute Python Script?  
(a) F5 (b) F2  
(c) F1 (d) F3
6. Which statement is used to skip the remaining part of the loop and start with next iteration?  
(a) break (b) pass  
(c) continue (d) null
7. Which of the following keyword is used to begin the function block?  
(a) define (b) def  
(c) finally (d) for
8. In Python, which operator is used to display a string in multiple number of times?  
(a) \* (multiplication) (b) + (addition)  
(c) -(subtraction) (d) / (division)
9. Marks = [20, 40, 60, 80, 100]  
print(Marks[-2])  
What will be the output?  
(a) 60 (b) 100 (c) 40 (d) 80
10. In Python, which of the following class declaration is correct?  
(a) class class\_name :  
(b) class class\_name < >  
(c) class class\_name  
(d) class class\_name [ ]
11. The Relational Database model was first proposed by :  
(a) C.D. Darween (b) Chris Date  
(c) E.F. Codd (d) Hugh Darween
12. Which is a Data Control Language command in SQL?  
(a) Alter (b) Grant  
(c) Truncate (d) Commit

[251]

13. CSV is expanded as :  
(a) Comma Separated Values  
(b) Condition Separated Values  
(c) Comma Special Values  
(d) Condition Special Values
14. Which of the following is not a scripting language?  
(a) Ruby (b) DBMS  
(c) Perl (d) JavaScript
15. Which SQL function returns the number of rows in a table?  
(a) SUM( ) (b) MAX( )  
(c) CHECK( ) (d) COUNT( )

### PART - II

Answer any six from the following questions.  
Question No. 24 is compulsory.  $6 \times 2 = 12$

16. List the characteristics of interface.
17. What are the characteristics of modules?
18. What are tokens in Python? List the types.
19. What is the use of replace() in Python? Write the general format of replace().
20. Write the syntax to create list with suitable example in Python.
21. What are the advantages of DBMS?
22. What are the two ways to read a CSV file in Python?
23. List the types of visualizations in Matplotlib.
24. What are the advantages of user-defined functions?

### PART - III

Answer any six from the following questions.  
Question No. 33 is compulsory.  $6 \times 3 = 18$

25. (a) What is selector?  
(b) What are the parts of a program?
26. Write note on Dynamic Programming.
27. Write the syntax of Nested if..elif...else statement with example.

28. Write short note on public and private data members in Python.
29. Write note on types of DBMS users.
30. Write short note on DELETE, TRUNCATE, DROP command in SQL.
31. What are the applications of scripting languages?
32. Write short note on : (a) GROUP BY (b) ORDER BY clause - in SQL.
33. What will be the output of the given Python program?  
a = "Computer"  
b = "Science"  
x = a[:4] + b[len(b)-3:]  
print(x)

### PART - IV

Answer the following questions:  $5 \times 5 = 25$

34. (a) Explain the types of variable scope.  
(OR)  
(b) Explain data types in Python with suitable example.

35. (a) Write the output for the following Python programs.

(i) j = 15  
while(j>=10):  
    print(j, end='t')  
    j=j-1  
else:  
    print("\t End of the loop")

(ii) k = 5.  
while(k<=9):  
    for i in range (1, k):  
        print(i, end='t')  
    print(end='n')  
    k=k+1

(OR)



Call @  
**9600175757**  
**8124301000**  
 orders@surabooks.com

www.kalviexpress.in

Buy Online @  
  
**surabooks.com**