SURA'S

COMPUTER SCIENCE

12th Standard

FREE
Practice Workbook
with
Lab Manual

Based on the Updated New Textbook

Salient Features

- Answers for all Textual Questions.
- 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.
  - (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 & May 2022 (Mar-2020 & May-2022), Supplementary Exam September - 2020, August 2021 & July-2022 (Sep-2020, Aug-2021 & July-'22) and First & Second Revision Test - 2022 (FRT-'22 & SRT-'22) questions are incorporated in the appropriate sections.
- Instant Supplementary Exam July 2022 Question Paper is given with answers.

SURA PUBLICATIONS
Chennai

2023-24 Edition

ISBN : 978-93-5330-523-9

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:

Sura Publications
1620, 'J' Block, 16th Main Road,
Anna Nagar, **Chennai - 600 040.**
Phones : 044 - 4862 9977, 044 - 4862 7755.
e-mail : orders@surabooks.com
website : www.surabooks.com

For Orders Contact

☎ 80562 94222
81242 01000
81243 01000
96001 75757
78718 02000
98409 26027

15/11/2022

TO ORDER WITH US

SCHOOLS and TEACHERS:

We are grateful for your support and patronage to **'SURA PUBLICATIONS'**
Kindly prepare your order in your School letterhead and send it to us.
For Orders contact: 81242 01000 / 81243 01000

DIRECT DEPOSIT

A/c Name	: **Sura Publications**
Our A/c No.	: **36550290536**
Bank Name	: **STATE BANK OF INDIA**
Bank Branch	: Padi
IFSC	: SBIN0005083

A/c Name	: **Sura Publications**
Our A/c No.	: **21000210001240**
Bank Name	: **UCO BANK**
Bank Branch	: Anna Nagar West
IFSC	: UCBA0002100

A/c Name	: **Sura Publications**
Our A/c No.	: **6502699356**
Bank Name	: **INDIAN BANK**
Bank Branch	: Asiad Colony
IFSC	: IDIB000A098

A/c Name	: **Sura Publications**
Our A/c No.	: **1154135000017684**
Bank Name	: **KVB BANK**
Bank Branch	: Anna Nagar
IFSC	: KVBL0001154

A/c Name	: **Sura Publications**
Our A/c No.	: **13240200032412**
Bank Name	: **FEDERAL BANK**
Bank Branch	: Anna Nagar
IFSC	: FDRL0001324

A/c Name	: **Sura Publications**
Our A/c No.	: **50200031530945**
Bank Name	: **HDFC BANK**
Bank Branch	: Cenotaph Road, Teynampet
IFSC	: HDFC0001216

A/c Name	: **Sura Publications**
Our A/c No.	: **446205000010**
Bank Name	: **ICICI BANK**
Bank Branch	: Anna Nagar
IFSC	: ICIC0004462

After Deposit, please send challan and order to our address.
email to : orders@surabooks.com / Whatsapp : 81242 01000.

G Pay **For Google Pay :**
98409 26027

For PhonePe :
98409 26027

DEMAND DRAFT / CHEQUE

Please send Demand Draft / cheque in favour of **'SURA PUBLICATIONS'** payable at **Chennai**. The Demand Draft / cheque should be sent with your order in School letterhead.

STUDENTS :

Order via Money Order (M/O) to

SURA PUBLICATIONS
1620, 'J' Block, 16th Main Road, Anna Nagar, Chennai - 600 040.
Phones : 044-4862 9977, 044-4862 7755.
Mobile : 96001 75757 / 81242 01000 / 81243 01000.
email : orders@surabooks.com Website : www.surabooks.com

CONTENTS

UNIT-I PROBLEM SOLVING TECHNIQUES

CHAPTER 1 FUNCTION

CHAPTER SNAPSHOT

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 *[Aug-2021; FRT-'22]*

(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? *[July-'22]*

(a) Subroutines (b) Function
(c) Files (d) Modules

[Ans. (b) Function]

3. Which of the following is a distinct syntactic block? *[PTA-6; FRT & May-'22]*

(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; FRT-'22]*

(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? *[PTA-4; FRT-'22]*

(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.** *[Aug-2021; FRT-'22]*

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 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. For example: strlen(), sqrt()	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; May-'22]*

(i) Parameter without Type *[FRT-'22]*

(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>=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, shows that as per the **typing** rule the entire if expression has type **int**.
- Since the if expression is of 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
```

i) Name of the function

ii) Identify the statement which tells it is a recursive function

iii) Name of the argument variable

iv) Statement which invoke the function recursively

v) 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 :**

(i) Pure functions are functions which will give exact result when the same arguments are passed.

(ii) 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.

(iii) 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. Lt'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 randomnumber:=
     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.

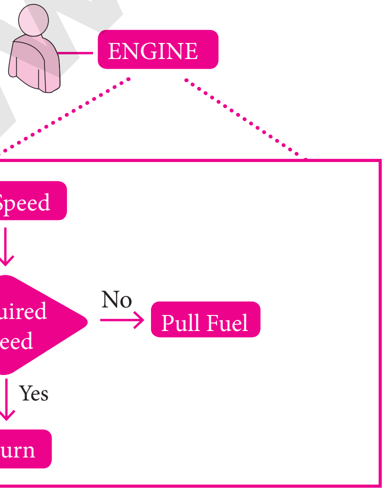
Implementation :

(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 2 DATA ABSTRACTION

CHAPTER SNAPSHOT

Evaluation

Part - I

Choose the best answer (1 Mark)

1. **Which of the following functions that build the abstract data type ?** *[Sep-2020; Aug-2021; July-'22]*
(a) Constructors (b) Destructors
(c) Recursive (d) Nested
[Ans. (a) Constructors]

2. **Which of the following functions that retrieve information from the data type?** *[FRT & May-'22]*
(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?** *[Aug-2021; FRT & May-'22]*

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; July-'22]*

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 :** lst = [(0,10), (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 datetype
(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; FRT-22]*

(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 become10 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; FRT-'22]*

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 and Selectors**

Constructors :

(i) Constructors are functions that build the abstract data type.

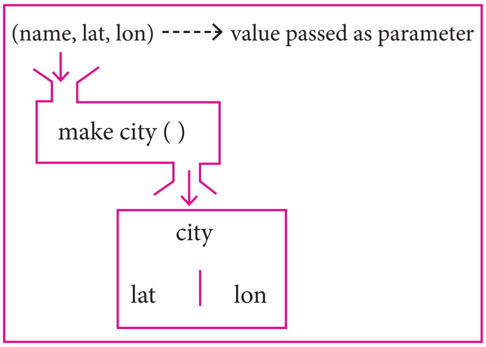
(ii) Constructors create an object, bundling together different pieces of information.

(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.



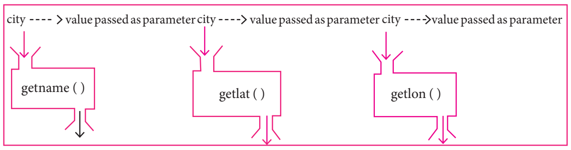
Constructor

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 used functions like
getname(city)
getlat(city)
getlon(city)
These are the selectors because these functions extract the information of the city object.


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

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
```

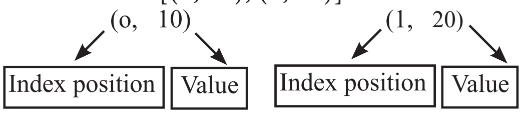
(iii) In the above example x will become10 and y will become 20. A second method for accessing the elements in a list is by the element selection operator, also expressed using square brackets.

(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.

lst[(0, 10), (1, 20)] - where



Pair :

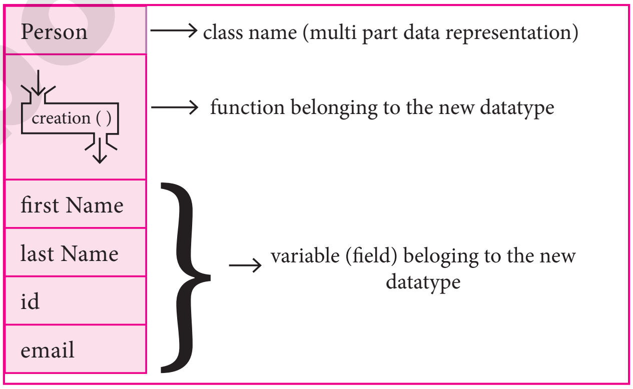
Any way of bundling two values together into one can be considered as a pair. Lists are a common method to do so. Therefore List can be called as Pairs.

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

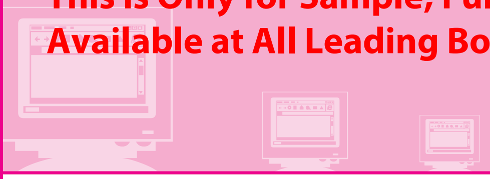
Ans. (i) The structure construct (In OOP languages it's called class construct) is used to represent multi-part objects where each part is named (given a name). Consider the following pseudo code:

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

The new data type Person is pictorially represented as


Person
class name (multi part data representation)
creation ( )
function belonging to the new datatype
first Name
last Name
id
email
variable (field) beloging to the new datatype

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="compsci@gamil.com"	setting a filed called email with value compsci@gmail.com
- - output of firstName : Padmashri	



Chapter 3

SCOPING

CHAPTER SNAPSHOT

Evaluation

Part - I

Choose the best answer (1 mark)

1. Which of the following refers to the visibility of variables in one part of a program to another part of the same program. *[FRT-'22]*

(a) Scope (b) Memory
(c) Address (d) Accessibility

[Ans. (a) Scope]

2. The process of binding a variable name with an object is called *[Sep-2020; Aug-2021; FRT-'22]*

(a) Scope (b) Mapping
(c) late binding (d) early binding

[Ans. (b) Mapping]

3. 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) =]

4. Containers for mapping names of variables to objects is called *[QY-2019; May-'22]*

(a) Scope (b) Mapping
(c) Binding (d) Namespaces

[Ans. (d) Namespaces]

5. Which scope refers to variables defined in current function? *[FRT & July-'22]*

(a) Local Scope (b) Global scope
(c) Module scope (d) Function Scope

[Ans. (a) Local Scope]

6. 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]

7. 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]

8. 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]

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. *[FRT-'22]*

Ans. The scope should be used for variables because; it limits a variables scope to a single definition. That is the variables are visible only to that part of the code. 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; May-'22]*

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; FRT & July-'22]*

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.

Example: self_n2 = n2

Part - III

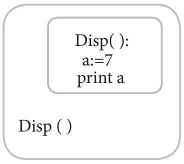
Answer the following questions (3 Marks)

1. Define Local scope with an example. *[Aug-2021]*

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 :

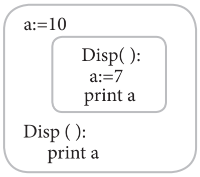
1. Disp(): 2. a:=7 3. print a 4. Disp()	Entire program 	Output of the Program 7

(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; FRT-'22]*

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  a:=10 Disp( ): a:=7 print a Disp ( ): print a	Output of the Program 7 10

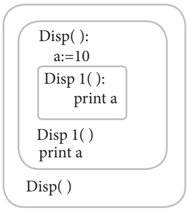
(iii) On execution of the above code the variable a which is defined inside the function displays the value 7 for the function call Disp() and then it displays 10, because a is defined in global scope.

3. Define Enclosed scope with an example. *[PTA-3; FRT-'22]*

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'
    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; May-'22]*

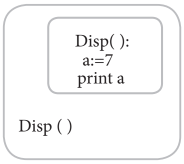
Ans. **Types of Variable Scope :**

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

Local Scope : *[FRT-'22]*

(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

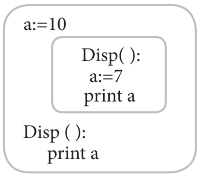
1. Disp(): 2. a:=7 3. print a 4. Disp()	Entire program 	Output of the Program 7

(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  a:=10 Disp( ): a:=7 print a Disp ( ): print a	Output of the Program 7 10

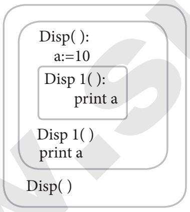
(iii) On execution of the above code the variable 'a' which is defined inside the function displays the value 7 for the function call Disp() and then it displays 10, because a is defined in global scope.

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

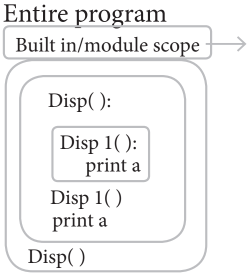
1. Disp():
2. a:=10
3. Disp1():
4. print a
5. Disp1()
6. print a
7. Disp()
	Entire program
	 Disp( ):
	a:=10
	Disp 1( ):
	print a
	Disp 1( )
	print a
	Disp( )
		Output of the Program
		10
		10

(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 : *[FRT-'22]*

(i) The built-in scope has all the names that are pre-loaded into the program scope when we start the compiler or interpreter.

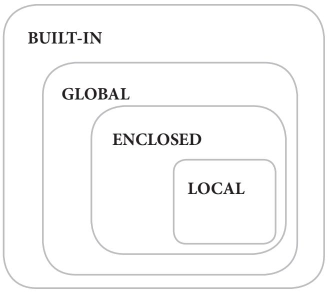
(ii) Any variable or function which is defined in the modules of a programming language has Built-in or module scope. Consider the following example.

Entire program  Built in/module scope Disp( ): Disp 1( ): print a Disp 1( ) print a Disp( )	Library files associated with the software

LEGB rule : *[May-'22]*

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.

3. Write any five benefits in using modular programming. *[Govt. MQP-2019]*

Ans. (i) Less code to be written.

(ii) A single procedure can be developed for reuse, eliminating the need to retype the code many times.

(iii) Programs can be designed more easily because a small team deals with only a small part of the entire code.

(iv) Modular programming allows many programmers to collaborate on the same application.

(v) The code is stored across multiple files.

(vi) Code is short, simple and easy to understand.

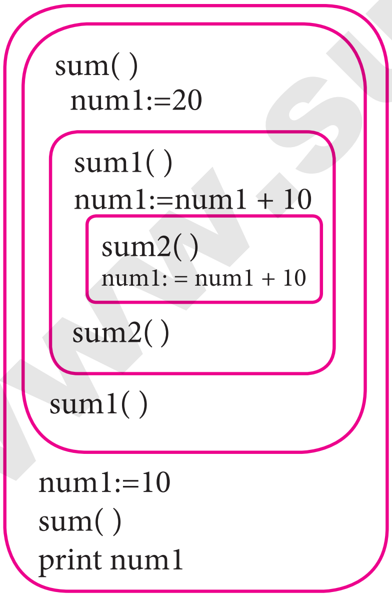
(vii) Errors can easily be identified, as they are localized to a subroutine or function.

(viii) The same code can be used in many applications.

(ix) The scoping of variables can easily be controlled.

Hands on Practice

1. Observe the following diagram and Write the pseudo code for the following.



```
sum( )
  num1:=20
    sum1( )
    num1:=num1 + 10
      sum2( )
      num1: = num1 + 10
    sum2( )
  sum1( )
num1:=10
sum( )
print num1
```

Ans. sum():

```
        num 1:=20
         sum1()
             num1:= num1 + 10
             sum2()
                 num1:= num1 + 10
              sum2()
             sum1()
             num1:= 10
             sum()
             Print num 1
```

PTA Questions and Answers

1 MARK

1. A variable which is declared inside a function which contains another function definition : *[PTA-1]*

(a) Local	(b) Global
(c) Enclosed	(d) Built-in

[Ans. (c) Enclosed]

2. Which are loaded as soon as the library files are imported to the program? *[PTA-3]*

(a) Built-in scope variables

(b) Enclosed scope variables

(c) Global scope variables

(d) Local scope variables

[Ans. (a) Built-in scope variables]

3. Which of the following is not the example of modules? *[PTA-5]*

(a) procedures	(b) subroutines
(c) class	(d) functions

[Ans. (c) class]

2 MARKS

1. What are modules? *[PTA-4]*

Ans. A module is a part of a program. Programs are composed of one or more independently developed modules.

Government Exam Questions and Answers

1 MARK

1. The kind of scope of the variable 'a' used in the pseudo code given below. *[Govt. MQP-2019]*

(a) Disp():	(b) a: = 7
(c) print a	(d) Disp()
(a) Local	(b) Global
(c) Enclosed	(d) Built-in

[Ans. (a) Local]

CHAPTER 4

ALGORITHMIC STRATEGIES

CHAPTER SNAPSHOT

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 Khowarizmi is called? *[PTA-6; Aug-2021; May-'22]*

(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? *[FRT-'22]*

(a) Bubble sort (b) Insertion sort
(c) Selection sort (d) All the above

[Ans. (c) 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(n2) (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) θ (n) (b) θ (nlogn)
(c) θ (n2) (d) θ (n(logn) 2)

[Ans. (a) θ (n)]

8. The Θ 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 subporblems]

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; Aug-2021]*

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. What is Insertion sort?

Ans. (i) Insertion sort is a simple sorting algorithm.

(ii) It works by taking elements from the list one by one and inserting then in their correct position in to a new sorted list.

(iii) This algorithm builds the final sorted array at the end.

4. What is Sorting? *[FRT-'22]*

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; May-'22]

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 *[Aug-2021; May-'22]*

(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 Ω :** 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 Θ :** When an algorithm has a complexity with lower bound = upper bound, say that an algorithm has a complexity O (n log n) and Ω (n log n), it's actually has the complexity Θ (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.

Chapter 6 CONTROL STRUCTURES

CHAPTER SNAPSHOT

Evaluation

Part - I

Choose the best answer (1 mark)

1. How many important control structures are there in Python? *[SRT-'22]*

(a) 3 (b) 4 (c) 5 (d) 6

[Ans. (a) 3]

2. elif can be considered to be abbreviation of *[May-'22]*

(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? *[Aug-2021]*

(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? *[July-'22]*

(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. *[Aug-2021]*

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 :** *[SRT-'22]*

```
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; July-'22]*

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

The syntax of range() is as follows:

range (start,stop,[step])

Where,

start – refers to the initial value

stop – refers to the final value

step – refers to increment value, this is optional part.

Part - III

Answer the following questions (3 marks)

1. Write a program to display *[PTA-5; May-'22]*

```
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 a
    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; SRT & July-'22]*

Ans. **Syntax:**

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

5. List the differences between break and continue statements. *[HY-2019; May-'22]*

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.
Syntax : break	**Syntax :** continue

Part - IV

Answer the following questions (5 marks)

1. Write a detail note on for loop. *[Govt. MQP-2019; Aug-2021; SRT, May & July-'22]*

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

(ii) **Syntax:**

```
for counter_variable in sequence:
        statements-block 1
[else: # optional block
        statements-block 2]
```

(iii) The counter_variable mentioned in the syntax is similar to the control variable that we used in the **for** loop of C++ and the sequence refers to the initial, final and increment value. Usually in Python, **for** loop uses the range() function in the sequence to specify the initial, final and increment values. range() generates a list of values starting from **start** till **stop-1**.

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

range (start,stop,[step])

Where,

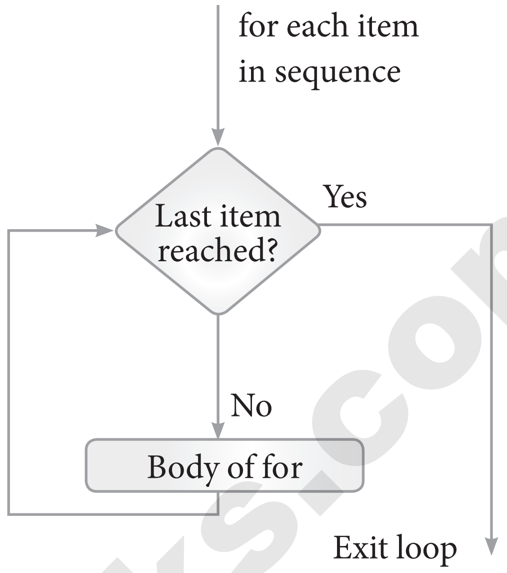
start – refers to the initial value

stop – refers to the final value

step – refers to increment value, this is optional part.

Examples for range() :

range (1,30,1)	- will start the range of values from 1 and end at 29
range (2,30,2)	- will start the range of values from 2 and end at 28
range (30,3,-3)	- will start the range of values from 30 and end at 6
range (20)	- will consider this value 20 as the end value(or upper limit) and starts the range count from 0 to 19 (remember always range() will work till stop -1 value only)



for loop execution

Example :

```
#Program to illustrate the use of for loop - to print single digit even number
for i in range (2,10,2):
    print (i, end=' ')
```

Output :

```
2 4 6 8
```

2. Write a detail note on if..else..elif statement with suitable example. *[HY-2019; Sep-2020]*

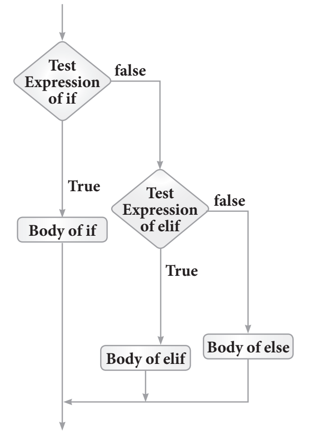
Ans. **Nested if..elif...else statement :**

(i) When we need to construct a chain of if statement(s) then **'elif'** clause can be used instead of **'else'**.

(ii) **Syntax :**

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

(iii) In the syntax of **if..elif..else** mentioned above, condition-1 is tested if it is true then statements-block1 is executed, otherwise the control checks condition-2, if it is true statements-block2 is executed and even if it fails statements-block n mentioned in **else** part is executed.



if..elif..else statement execution

(iv) **'elif'** clause combines **if..else-if..else** statements to one **if..elif…else. 'elif'** can be considered to be abbreviation of **'else if'**. In an **'if'** statement there is no limit of 'elif' clause that can be used, but an **'else'** clause if used should be placed at the end.

(v) **Example :** #Program to illustrate the use of nested if statement

Average	Grade
>=80 and above	A
>=70 and <80	B
>=60 and <70	C
>=50 and <60	D
Otherwise	E

```
m1=int (input("Enter mark in first subject : "))
m2=int (input("Enter mark in second subject : "))
avg= (m1+m2)/2
if avg>=80:
    print ("Grade : A")
elif avg>=70 and avg<80:
    print ("Grade : B")
elif avg>=60 and avg<70:
    print ("Grade : C")
elif avg>=50 and avg<60:
    print ("Grade : D")
else:
    print ("Grade : E")
```

Output 1:

```
Enter mark in first subject : 34
Enter mark in second subject : 78
Grade : D
```

Output 2 :

```
Enter mark in first subject : 67
Enter mark in second subject : 73
Grade : B
```

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

Ans.

```
for a in range (100, 1000):
    if a %2==1:
        print b
```

Output :

```
101, 103, 105, 107 ... .... 997, 999
```

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

Ans. **Multiplication table :**

```
num = int(input("Enter the number : "))
prit("multiplication Table of", num)
for i in range(1,11):
    print (num, "x", i, " = ", num*i)
```

Output :

```
Enter the number : 6
Multiplication Table of 6
    6 × 1 = 6
    6 × 2 = 12
    6 × 3 = 18
    6 × 4 = 24
    6 × 5 = 30
    6 × 6 = 36
    6 × 7 = 42
    6 × 8 = 48
    6 × 9 = 54
    6 × 10 = 60
```

Hands on Experience

1. Write a program to check whether the given character is a vowel or not. *[QY-2019]*

Ans. **Program :**

```
ch = input ("Enter a character")
if ch in ('a', 'A', 'e', 'E', 'i', 'I','o','O','u', 'U'):
print (ch, 'is a vowel')
else :
print (ch, 'the letter is not a vowel')
```

Chapter 8

STRINGS AND STRING MANIPULATION

CHAPTER SNAPSHOT

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] = "-"~ *[May-'22]*

(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? *[SRT-'22]*

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

[Ans. (a) +]

4. Defining strings within triple quotes allows creating: *[HY-2019]*

(a) Single line Strings (b) Multiline Strings

(c) Double line Strings (d) Multiple Strings

[Ans. (b) Multiline Strings]

5. Strings in python: *[Aug-2021]*

(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; July-'22]*

(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? *[Aug-2021; SRT-'22]*

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? *[July-'22]*

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

Ans. **Output :** School School School

5. What is slicing? *[PTA-6; Aug-2021]*

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]*

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

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())
		Chennai

(b) swapcase( )	It will change case of every character to its opposite case vice-versa.	>>> str1="tAmiL NaDu"
		>>> print(str1.swapcase())
		TaMIl nAdU

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.	>>> str1="Raja Raja Chozhan"
		>>> print(str1. count('Raja'))
		2
		>>> print(str1. count('r'))
		0
		>>> print(str1. count('R'))
		2
		>>> print(str1. count('a'))
		5
		>>> print(str1. count('a',0,5))
		2
		>>> print(str1. count('a',11))
		1

Part - IV

Answer the following questions (5 Marks)

1. Explain about string operators in python with suitable example. *[PTA-2; HY-2019; SRT-'22]*

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
```

Chapter 10

PYTHON CLASSES AND OBJECTS

CHAPTER SNAPSHOT

Evaluation

Part - I

Choose the best answer (1 mark)

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

(a) Constructor and Classes
(b) Constructor and Object
(c) Classes and Objects
(d) Constructor and Destructor

[Ans. (c) Classes and Objects]

2. Functions defined inside a class:

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

[Ans. (c) Methods]

3. Class members are accessed through which operator? *[Mar.-2020; SRT & July-'22]*

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

[Ans. (b) .]

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

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

[Ans. (d) __init__( )]

5. A private class variable is prefixed with

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

[Ans. (a) __]

6. Which of the following method is used as destructor? *[PTA-1; QY-2019; May-'22]*

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

[Ans. (d) __del__( )]

7. Which of the following class declaration is correct? *[PTA-6; Sep-2020]*

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

[Ans. (c) class class_name:]

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

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

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

[Ans. (b) Tamil]

9. Which of the following is the private class variable? *[PTA-2]*

(a) __num (b) ##num

(c) $$num (d) &&num

[Ans. (a) __num]

10. The process of creating an object is called as: *[HY-2019; Aug-2021]*

(a) Constructor (b) Destructor

(c) Initialize (d) Instantiation

[Ans. (d) Instantiation]

Part - II

Answer the following questions (2 Marks)

1. What is class? *[PTA-1]*

Ans. (i) Class is the main building block in Python.

(ii) Class is a template for the object.

(iii) Object is a collection of data and function that act on those data.

(iv) Objects are also called as instances of a class.

2. What is instantiation? *[PTA-6; SRT-'22]*

Ans. Once a class is created, next to create an object or instance of that class. The process of creating object is called as "Class Instantiation".

Syntax :

```
Object_name = class_name( )
```

3. What is the output of the following program?

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

Ans. Output :

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

4. How will you create constructor in Python?

Ans. (i) **"init"** is a special function begin an end with double underscore in Python act as a Constructor.

(ii) Constructor function will automatically executed when an object of a class is created.

General format of __init__ method (Constructor function)

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

5. What is the purpose of Destructor? *[PTA-2; SRT-'22]*

Ans. (i) Destructor is also a special method gets executed automatically when an object exit from the scope.

(ii) In Python,_del_() method is used as destructor.

General format :

```
    def_del_(self):
    <statements>
```

Part - III

Answer the following questions (3 Marks)

1. What are class members? How do you define it? *[PTA-1; Aug-2021]*

Ans. Variables defined inside a class are called as "Class Variable" and functions are called as "Methods". Class variable and methods are together known as members of the class. The class members should be accessed through objects or instance of class. A class can be defined anywhere in a Python program.

Syntax for Defining a Class :

```
class class_name :
        statement_1
        statement_2
        ....................
        ....................
        statement_n
```

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

Ans. Code :

```
class Sample :
    def_init_(self, n1, n2):
        self._n1=n1
        self._n2=n2
def sum(self):
    print ("Class Variable 1:",self._n1)
    print ("Class Variable 2:", self._n2)
S=Sample (5,10)
S.sum()
```

Output :

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

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

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

Output :

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

Ans. In line No.8, del F.display will not come.

4. What is the output of the following program?

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

[July-'22]

Ans. Good Morning Bindu Madhavan

5. How to define constructor and destructor in Python? *[PTA-4; Mar-2020; Sep-2020]*

Ans. **Constructor :**

(i) Constructor is the special function that is automatically executed when an object of a class is created. In Python, there is a special function called “init” which act as a Constructor.

(ii) It must begin and end with double underscore.

(iii) Constructor function will automatically executed when an object of a class is created.

General format of constructor :

```
    def__init__(self, [args ................]):
    <statements>
```

Example : Program to illustrate Constructor

```
class Sample:
    def __init__(self, num):
    print("Constructor of class Sample...")
    self.num=num
    print("The value is :", num)
S=Sample(10)
```

Destructor :

(i) Destructor is also a special method gets executed automatically when an object exit from the scope.

(ii) In Python, __del__() method is used as destructor.

General format of constructor :

```
    def__del__(self):
    <statements>
```

Example : Program to illustrate about the __del__( ) method

```
class Sample:
    num=0
    def __init__(self, var):
        Sample.num+=1
        self.var=var
        print("The object value is = ", var)
        print("The value of class variable is=
        ", Sample.num)
    def __del__(self):
        Sample.num-=1
        print("Object with value %d is exit
        from the scope"%self.var)
S1=Sample(15)
S2=Sample(35)
S3=Sample(45)
```

Part - IV

Answer the following questions (5 marks)

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

Ans. **Code :**

```
stationary = {}
print("\n1.Add Item \n2. Delete item \n3.Exit")
ch=int(input("\nEnter your choice:"))
while(ch==1) or (ch==2):
    if(ch==1)
        n=int(input("\nEnter the Number of
        Items to be added in the Dictionary:"))
        for i in range(n):
    item=input("\nEnter an Item Name:")
    brand=input("\nEnter the Brand Name:")
    stationary[item]=brand
print(stationary)
elif(ch==2):
    ritem=input("\nEnter the item to be
            removed from the Dictionary:")
    stationary.pop(ritem)
    print(stationay)
ch=int(input("\nEnter your choice:"))
```

Output :

```
1.  Add Item
2.  Delete item
3.  Exit
Enter your choice : 1
Enter the Number of Items to be added in the
                          stationary shop : 2
Enter an Item Name : Pen
Enter the Brand Name : Rorito
Enter an Item Name : Pencil
Enter the Brand Name : Camlin
{'Pen' : 'Rorito', 'Pencil' : 'Camlin'}
Enter your choice : 2
Enter the item to be removed from the Dictionary
                                        : Pen
{'Pencil': 'Camlin'}
Enter your choice : 3
```

Hands on Experience

1. Write a program using class to store name and marks of students in list and print total marks.

Ans. Class stud :

```
Class stud :
def__init__(self):
    self.name=" "
    self.m1=0
    self.m2=0
    self.tot=0
def gdata(self):
    self.name = input("Enter your name")
    self.m1 = int(input("Enter marks 1"))
    self.m2 = int(input("Enter marks 2"))
    self.tot = self.m1 + self.m2
def disp(self):
    print(self.name)
    print(self.m1)
    print(self.m2)
    print(self.tot)
mlist = []
st = stud()
st. gdata()
mlist.append(st)
for x in mlist :
    x. disp()
```

Output :

```
    Enter your name      Ram
    Enter marks 1        100
    Enter marks 2        100
Ram      100      100    200
```

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

Ans. Class Tr:

```
Class Tr:
def__init__(self, a, b, c):
    self.a = float(a)
    slef.b = float(b)
    self.c = float(c)
def area(self):
    s = (self.a + self.b + self.c)/2
    return((s*(S.self.a)*(s.self.b)*(s.self.c)**0.5)
a = input("Enter side 1 :")
b = input("Enter side 2 :")
c = input("Enter side 3 :")
ans = Tr(a,b,c)
print(ans.area())
```

Output :

```
    Enter side 1   :     3
    Enter side 2   :     4
    Enter side 3   :     5
        6.0
```

3. Write a menu driven program to read, display, add and subtract two distances.

Ans.

```
class Dist :
def__init__(self):
     self.dist 1=0
     self.dist 2=0
def read(self):
     self.dist 1 =int(input("Enter distance 1"))
     sefl.dist 2 =int(input("Enter distance 2"))
def disp(self):
     print("distance 1", self.dist 1)
     print("distance 2", self.dist 2)
def add(self):
     print("Total distance", self.dist 1 + self.dist 2)
def sub(self):
     print("Subtracted distance", self.dist 1-self.dist 2)
d=Dist()
choi = "y"
while(choi =="y"):
     print("1. accept\n2. Display \n3. Total \n4. Subtract")
ch = int(input("Enter your choice"))
if(ch==1):
     d.read()
elif(ch==2)
     d.disp()
elif(ch==3):
     d.add()
elif(ch==4):
     d.sub()
else:
     print("Invalid Input ...")
choi = input("Do you want to continue")
```

Output :

```
     1. Accept
     2. Display
     3. Add
     4. Subtract
     Enter your choice : 1
          Enter distance 1 : 100
          Enter distance 2 : 75
     Do you want to continue .. y
          1. Accept
          2. Display
          3. Add
          4. Subtract
     Enter your choice : 3
          Total distances : 175
     Do you want to continue ..y
     1. Accept
     2. Display
     3. Add
     4. Subtract
     Enter your choice : 2
          Enter distance 1 : 100
          Enter distance 2 : 75
     Do you want to continue ..y
     1. Accept
     2. Display
     3. Add
     4. Sub
     Enter your choice : 4
          Subtracted distance : 25
     Do you want to continue .. N
```

PTA QUESTIONS AND ANSWERS

1 MARK

1. In Python the class method must have which named argument as first argument? ***[PTA-3]***

(a) self (b) rec

(c) global (d) key

[Ans. (a) self]

2. The function defined inside a class is called as ______. ***[PTA-4]***

(a) Attribute (b) Parameter

(c) Arguments (d) Methods

[Ans. (d) Methods]

3. The symbol of project in relational algebra of DBMS : ***[PTA-5]***

(a) σ (b) Π (c) ∩ (d) ∪

[Ans. (b) Π]

2 MARKS

1. Write the syntax of class instantiation. ***[PTA-5]***

Ans. **Syntax :**

object_name = class_name()

Note that the class instantiation uses function notation. ie.class_name with.

2. Write the general format of slicing operation. ***[PTA-6]***

Ans. **General format of slice operation:**

str[start:end]

3 MARKS

1. **What is Public and Private data member in Python?** *[PTA-3; Sep-2020; SRT-'22]*

Ans. (i) The variables which are defined inside the class is public by default. These variables can be accessed anywhere in the program using dot operator.

(ii) A variable prefixed with double underscore becomes private in nature. These variables can be accessed only within the class.

5 MARKS

1. **Find the output of the following Python code** *[PTA-1]*

```
class Sample:
    num=0
    def __init__(self, var):
            Sample.num+=1
            self.var=var
            print("The object value is = ", var)
            print("The count of object created = ", Sample.num)
S1=Sample(15)
S2=Sample(35)
S3=Sample(45)
```

Ans. In the program, class variable **num** is shared by all three objects of the class Sample. It is initialized to zero and each time an object is created, the num is incremented by 1. Since, the variable shared by all objects, change made to num by one object is reflected in other objects as well.

2. **What will be the output of the following Python code?** *[PTA-2]*

```
class String
    def__init__(self):
            self.uppercase=0
            self.lowercase=0
            self.vowels=0
            self.consonants=0
            self.spaces=0
            self.string=""
    def getstr(self):
            self.string= str(input("Enter a String: "))
    def count_upper(self):
            for ch in self.string:
                if (ch.isupper()):
                self.uppercase+=1
    def count_lower(self):
            for ch in self.string:
                if (ch.islower()):
                self.lowercase+=1
    def count_vowels(self):
            for ch in self.string:
                if (ch in ('A', 'a', 'e', 'E', 'i', 'I', 'o', 'O', 'u', 'U')):
                self.vowels+=1
    def count_consonants(self):
        v=('A', 'a', 'e', 'E', 'i', 'I', 'o', 'O', 'u', 'U')
        for ch in self.string:
            if ch not in v and ch.isalpha():
                self.consonants+=1
    def count_space(self):
            for ch in self.string:
                if (ch==" "):
                self.spaces+=1
    def execute(self):
            self.count_upper()
            self.count_lower()
            self.count_vowels()
            self.count_consonants()
            self.count_space()
    def display(self):
            print("The given string contains...")
            print("%d Uppercase letters"%self.uppercase)
            print("%d Lowercase letters"%self.lowercase)
            print(%d Vowels"%self.vowels)
            print("%d Consonants"%self.consonants)
            print("%d Spaces"%self.spaces)
S = String()
S.getstr()
S.execute()
S.display()
```

Ans. Output :

```
Enter a string : Welcome to Learn Computer Science

The given string contains....
5  Uppercase letters
24 Lowercaase letters
12 Vowels
17 Consonants
3  Spaces
```

3. Rewrite the following Python program to get the given output: *[PTA-3]*

OUTPUT :

```
Enter Radius: 5
The area = 78.5
The circumference = 34.10
```

CODE :

```
Class circle( )
pi=3.14
def__init__(self, radius):
        self=radius
DEF area(SELF):
        Return
        Circle.pi + (self.radius * 2)
Def circumference (self):
Return 2*circle.pi * self.radius
r = input("Enter radius= ")
c = circle(r)
print "The Area: ", c.area( )
print("The circumference=", c)
```

Ans.

```
Class circle:
     pi=3.14
def__init__(self, radius):
     self.radius = radius
def area(self):
     return circle.pi*(self.radius**2)
def circumference(self):
     return2*circle.pi*self.radius
r=int(input("Enter Radius:"))
     c = circle(r)
print("The Area= ", c.area( ))
print("The circumference = ", c.circumference)
```

Government Exam Questions and Answers

1 MARK

1. A variable prefixed with double underscore is *[Govt. MQP-2019]*

(a) private
(b) public
(c) protected
(d) static

[Ans. (a) private]

3 MARKS

1. What is the output of the following program? *[Govt. MQP-2019]*

```
class Greeting:
     def__init__(self, name):
          self.__name = name
     def display(self):
          print("Good Morning", self.__name)
obj=Greeting('Tamil Nadu')
obj.display()
```

Ans. Good Morning Tamil Nadu

2. What is Constructor? *[QY-2019]*

Ans. Constructor is the special function that is automatically executed when an object of a class is created. In Python, there is a special function called "init" which act as a Constructor. It must begin and end with double underscore. This function will act as an ordinary function; but only difference is, it is executed automatically when the object is created. This constructor function can be defined with or without arguments. This method is used to initialize the class variables.

General format of __init__ method (Constructor function)

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

3. Write a Python program to check and print if the given number is odd or even using class.

Ans. *[HY-2019]*

```
class Odd_Even:
def check(self, num):
     if num%2==0:
          print(num," is Even number")
else:
          print(num," is Odd number")
n=Odd_Even()
x = int(input("Enter a value: "))
n.check(x)
```

4. What is the output of the following program? *[Govt. MQP-2019]*

```
class Greeting:
     def__init__(self, name):
          self.__name = name
     def display(self):
          print("Good   Morning",   self.__name)
     obj=Greeting('Tamil Nadu')
     obj.display()
```

Ans. Output :

Good Morning Tamil Nadu

5. **Write the output of the following program.** *[Mar-2020]*

```
class Hosting:
    def __init__(self, name):
            self.__name = name
    def display(self):
            print("Welcome to", self.__name)
obj=Hosting("Python Programming")
obj.display()
```

Ans. Output : Welcome to Python Programming.

6. **What will be the output of the following program?** *[SRT-'22]*

```
class stud:
m1, m2, m3 = 45, 91, 71
def process(self) :
   sum = stud.m1 + stud.m2 + stud.m3
   avg  =  sum/3
   print("Total Marks = ", sum)
   print("Average Marks = ", avg)
   return
s. stud( )
s. process( )
```

Ans.

```
Total Marks = 207
Average Marks = 69.0
```

5 MARKS

1. **How will you create the class, and objects in python.** *[QY-2019]*

(or)

How to define a class in Python? Explain with example. *[SRT-'22]*

Ans. (i) In Python, a class is defined by using the keyword class. Every class has a unique name followed by a colon (:).

Syntax:

```
class class_name:
        statement_1
        statement_2
        …………..
        …………..
        statement_n
```

(ii) Where, statement in a class definition may be a variable declaration, decision control, loop or even a function definition. Variables defined inside a class are called as "Class Variable" and functions are called as "Methods". Class variable and methods are together known as members of the class. The class members should be accessed through objects or instance of class. A class can be defined anywhere in a Python program.

Example:

```
class Sample:
    x, y = 10, 20          # class variables
```

In the above code, name of the class is Sample and it has two variables x and y having the initial value 10 and 20 respectively. To access the values defined inside the class, you need an object or instance of the class.

2. **What will be the output of the following program?** *[SRT-'22]*

```
Class Sample :
     num = 0
     def _ _init_ _(self, var):
         Sample.num + = 1
         self.var = var
         print("The object value is = ", var)
         print("The count of object created = ",
                                     Sample.num)
     S1 = Sample(15)
     S2 = Sample(35)
     S3 = Sample(45)
```

Ans.

```
The object value is = 15
The count of object created = 1
The object value is = 35
The count of object created = 2
The object value is = 45
The count of object created = 3
```

Additional Questions and Answers

Choose the Correct Answer 1 MARK

1. **Which of the following is not an object oriented language?**

(a) C (b) C++
(c) Java (d) Python

[Ans. (a) C]

2. **Which of the following is called as instances of a class ?**

(a) Methods (b) Objects
(c) Functions
(d) Datatypes **[Ans. (b) Objects]**

3. Functions of the class are called as

(a) Methods
(b) Members
(c) Variables
(d) Loog **[Ans. (a) Methods]**

4. In Python, every class name followed by

(a) ; (b) : (c) : : (d) .
[Ans. (b) :]

5. Which of the following with a valid class definition?

(a) Class classname () statement_1
(b) Class classname : : statement_1
(c) Class classname statement_1
(d) Class classname statement_1
[Ans. (c) Class classname statement_1]

6. Which of the following is valid syntax for crating objects?

(a) objectname = classname ()
(b) objectname : classname ()
(c) objectname = classname
(d) classname = Objectname ()
[Ans. (c) objectname = classname]

7. Which of the operator used to accessing members of the class?

(a) . (b) : (c) ; (d) ,
[Ans. (a) .]

8. Which of the following can be accessed by using object with dot (.) operator?

(a) List (b) Tuples
(c) Dictionary (d) None of these
[Ans. (d) None of these]

9. Which of the following is valid syntax of accessing class members

(a) objectname = classmember ()
(b) objectname . classmember ()
(c) objectname . classmember
(d) objectname . classmember
[Ans. (b) objectname . classmember ()]

10. Write the output for the following

```
class test
x, y = 10, 5
    s = test ()
    print (s. x + s. y)
```

(a) 10 (b) 5 (c) 15 (d) 105
[Ans. (c) 15]

11. Which position of the argument named self in python class method?

(a) First (b) Second
(c) Third (d) Last
[Ans. (a) First]

12. Which argument doesn't need a value when we call the method?

(a) this (b) self
(c) var (d) first
[Ans. (b) self]

13. Which of the following argument values automatically by python?

(a) self (b) this
(c) class (d) object
[Ans. (a) self]

14. How many argument can be taken by Python method even when a method is defined with one argument?

(a) 1 (b) 3 (c) 2 (d) 4
[Ans. (c) 2]

15. Which of the following is automatically executed when an object of a class is created?

(a) constructor (b) destructor
(c) class (d) members
[Ans. (a) constructor]

16. In Python, which function will act as a constructor?

(a) int (b) inti
(c) classname (d) init
[Ans. (d) init]

17. In Python, constructor must begin and end with

(a) _ _ and _ _ (b) _ _ and _ _
(c) + + and + + (d) + - and - +
[Ans. (a) _ _ and _ _]

18. Which of the following is used to initialize the class variables?

(a) Destructor (b) Object
(c) Constructor (d) Classmember
[Ans. (c) Constructor]

19. Which of the following gets executed automatically when an object exit from the scope?

(a) Destructor (b) Constructor
(c) Class (d) Object
[Ans. (a) Destructor]

20. By default, the class variables are

(a) Private (b) Public
(c) Protected (d) Method
[Ans. (b) Public]

21. **Which of the following is a valid private variable in python?**
(a) - i (b) i - (c) - - i (d) i - -
[Ans. (c) - - i]

22. **Which of the following variables can be accessed only within the class?**
(a) Protected (b) Public
(c) Private (d) None of these
[Ans. (d) None of these]

Choose and filling the blanks

1. **____ and ___ are the key features of object oriented programming.**
(a) List and tuples
(b) Set and dictionary
(c) Classes and objects
(d) Variables and methods
[Ans. (c) Classes and objects]

2. **____ is the main building block in python.**
(a) Objects (b) Methods
(c) Constructors (d) Class
[Ans. (d) Class]

3. **Class is a template for the _____**
(a) Method (b) Members
(c) Object (d) Destructor
[Ans. (c) Object]

4. **______ may be a variable declaration, decision control, loop or even a function definition.**
(a) Class members
(b) Class instantiation
(c) Class method
(d) Class definition [Ans. (d) Class definition]

5. **In Python, a class is defined by using the ____ class.**
(a) Operator (b) Identifier
(c) Object (d) Keyword
[Ans. (d) Keyword]

6. **Class variable and methods are together known as ______ of the class.**
(a) Objects (b) Functions
(c) Statements (d) Members
[Ans. (d) Members]

7. **The ______ of the class should be accessed through instance of a class.**
(a) Objects (b) Members
(c) Functions (d) Tuples
[Ans. (b) Members]

8. **The process of creating object is called as ____**
(a) Class definition (b) Class declaration
(c) Class instantiation (d) Class objects
[Ans. (c) Class instantiation]

9. **When class variable declared within class, methods must be prefixed by the _______ and ______**
(a) classnam, : (b) classname, .
(c) :, classname
(d) classname, objectname
[Ans. (b) classname, .]

10. **Constructor must begin and with double ______**
(a) Colon (b) Semicolon
(c) Dot (d) Underscore
[Ans. (d) Underscore]

11. **In Python, _____ method is used as destructor.**
(a) - - init - - () (b) - - des - - ()
(c) - - del - - ()
(d) - - destructor - - () [Ans. (b) - - des - - ()]

12. **A variable prefixed with _____ become private in nature.**
(a) double underscore (b) double colon
(c) double dot (d) double hyphen
[Ans. (a) double underscore]

Choose the correct statement

1. (a) objectname.classmember()
(b) objectname.classmember
(c) objectname().classmember
(d) objectname : classmember
[Ans. (b) objectname.classmember]

2. **Which of the following is correct declaration of constructor?**
(a) - - init - -
(b) - - init - - ()
(c) - - classname - - ()
(d) - - classname - - () [Ans. (b) - - init - - ()]

3. (a) objectname = classname
(b) objectname : classname ()
(c) objectname = classname ()
(d) objectname : : classname ()
[Ans. (c) objectname = classname ()]

Choose the incorrect statement

1. (i) The process of creating object is called "Class definition"
(ii) The class members are accessed using dot (.) operator.

(iii) The first argument of the class method is not self.

(iv) The method argument defined with one argument it takes two arguments with default.

(a) i and iv (b) ii and iii

(c) iv only (d) i and iii

[Ans. (d) i and iii]

2. (i) Constructor executed automatically when the object is created

(ii) In Python, "init" which act as a destructor.

(iii) In Python, constructor can be defined only with arguments.

(iv) Construct is used to initialize the class variables.

(a) i and iii (b) ii and iv

(c) iii, iv and ii (d) ii and iii

[Ans. (d) ii and iii]

Very Short Answers — 2 MARKS

1. Write the general form of declaring class in Python.

Ans. In Python, a class is defined by using the keyword class. Every class has a unique name followed by a colon (:).

Syntax :

class class_name:
statement_1
statement_2
…………..
…………..
statement_n

2. Write the syntax for the following
(i) Creating objects
(ii) Accessing class members

Ans. (i) Object – name = class_name ()

(ii) Object – name = class_member

3. Differentiate python class function and ordinary function.

Ans. Python class function or method is very similar to ordinary function with a small difference. The class method must have the first argument named as self.

4. Name the function which acts as a constructor and destructor.

Ans. Constructor -(- - init - - ( ))
Destructor -(- - del - - ( ))

5. Write a program in python that illustrate the use of constructor.

Ans. **Program to illustrate Constructor :**

```
class Sample:
    def __init__(self, num):
        print("Constructor of class Sample...")
        self.num=num
        print("The value is :", num)
S=Sample(10)
```

Short Answers — 3 MARKS

1. Write a note on object.

Ans. (i) Object is a collection of data and function that act on those data. Class is a template for the object.

(ii) According to the concept of Object Oriented Programming, objects are also called as instances of a class.

(iii) In Python, everything is an object. For example, all integer variables that we use in our program is an object of class int. Similarly all string variables are also object of class string.

2. Write a note on self argument used in python class function.

Ans. (i) Python class function or Method is very similar to ordinary function with a small difference that, the class method must have the first argument named as self.

(ii) No need to pass a value for this argument when we call the method. Python provides its value automatically.

(iii) Even if a method takes no arguments, it should be defined with the first argument called self.

(iv) If a method is defined to accept only one argument it will take it as two arguments ie. self and the defined argument.

3. Explain the working of the following program.

```
class Sample:
    def __init__(self, num):
        print("Constructor of class Sample...")
        self.num=num
        print("The value is :", num)
S=Sample(10)
```

Ans. (i) The above class “Sample”, has only a constructor with one argument named as num. When the constructor gets executed, first the print statement, prints the “Constructor of class Sample….”, then, the passing value to the constructor is assigned to self.num and finally it prints the value passed along with the given string.

(ii) The above constructor gets executed automatically, when an object S is created with actual parameter 10. Thus, the Python displays the following output.

(iii) Constructor of class Sample...

The value is : 10

Class variable defined within constructor keep count of number of objects created with the class.

4. Write a python program to find total and average marks using class.

Ans.
```
class Student:
        mark1, mark2, mark3 = 45, 91, 71
                                    #class variable
    def process(self):              #class method
        sum = Student.mark1 + Student.mark2 +
                                   Student.mark3
        avg = sum/3
        print("Total Marks = ", sum)
        print("Average Marks = ", avg)
        return
    S=Student()
    S.process()
```

5. Fill up the blanks in the following program to get the output :

Value of x = 10
Value of y = 20
Sum of x and y = 30

```
Class sample:
    x, _____ = 10, 20 ------------------①
    s = _______ ------------------------②
print (“value of x =”, _____) ------------③
print (“value of y =”, _____) ------------④
print (“sum of x and y =”, _____) --------⑤
```

Ans. 1. – y

2. – sample ()

3. – s. x

4. – s. y

5. – s. x + s. y

6. Read the following program. Answer the following question.

```
Class sample:
x, y = 10, 20
s = sample ()
print (s. x + s. y)
```

1. What does sample denotes?

2. What does x, y denotes?

3. What does s denotes?

Ans. 1. It denotes class name

2. x, y is a class variables of the class

3. S is an object created to access the members of the class

Long Answers **5 MARKS**

1. Write a program to check and print if the given number is negative or positive using class.

Ans.
```
class test:
def check (self, num)
    if num> 0:
            print (num, “is positive number”)
else:
            print (num, “is negative number”)
    n = test ()
    x = int (input(“Enter the number”))
    n. check (x)
```

2. Write a menu driven program that keeps record of books available in you school library.

Ans.
```
class Library:
def __init__(self):
            self.bookname=""
            self.author=""
def getdata(self):
            self.bookname = input("Enter Name
                                   of the Book: ")
            self.author = input("Enter Author of
                                      the Book: ")
def display(self):
    print("Name of the Book: ",self.bookname)
    print("Author of the Book: ",self.author)
    print(“\n”)
book=[] #empty list
ch = 'y'
while(ch=='y'):
```

```
          print("1. Add New Book \n 2.Display Books")
          resp = int(input("Enter your choice : "))
     if(resp==1):
          L=Library()
          L.getdata()
          book.append(L)
     elif(resp==2):
          for x in book:
               x.display()
     else:
          print("Invalid input....")
     ch = input("Do you want continue....")
```

3. Write a program to store product and its cost price. Display all the available products and prompt to enter quantity of all the products. Finally generate a bill which displays the total amount to be paid.

Ans.

```
     class MyStore:
          __prod_code=[]
          __prod_name=[]
          __cost_price=[]
          __prod_quant=[]
     def getdata(self):
          self.p = int(input("Enter no. of products you need to store: "))
          for x in range(self.p):
          self.__prod_code.
         append(int(input("Enter Product Code: ")))
          self.__prod_name.append(str(input("Enter Product Name: ")))
          self.__cost_price.append(int(input("Enter Cost price: ")))
     def display(self):
          print("Stock in Stores")
          print("-----------------------------------")
          print("Product Code \t Product Name \t Cost Price")
          print("-----------------------------------")
          for x in range(self.p):
          print(self.__prod_code[x], "\t\t", self.__prod_name[x], "\t\t", self.__cost_price[x])
          print("-----------------------------------")
     def print_bill(self):
          total_price = 0
          for x in range(self.p):
               q=int(input("Enter the quantify for the product code %d : "%self.__prod_code[x]))
     self.__prod_quant.append(q)
     total_price = total_price +self.__cost_price[x]*self.__prod_quant[x]
          print(" Invoice Receipt ")
          print("-----------------------------------")
          print("Product Code\t Product Name\t Cost Price\t Quantity \t Total Amount")
          print("-----------------------------------")
     for x in range(self.p):
          print(self.__prod_code[x], "\t\t", self.__prod_name[x], "\t\t",
               self.__cost_price[x], "\t\t", self.__prod_quant[x], "\t\t",
               self.__prod_quant[x]*self.__cost_price[x])
          print("-----------------------------------")
          print(" Total Amount = ", total_price)
```

☆☆☆

CHAPTER 13 PYTHON AND CSV FILES

CHAPTER SNAPSHOT

Evaluation

Part - I

Choose the best answer (1 mark)

1. A CSV file is also known as a .... *[Mar.-2020]*

(a) Flat File (b) 3D File
(c) String File (d) Random File

[Ans. (a) Flat File]

2. The expansion of CRLF is *[Govt. MQP-2019; May-'22]*

(a) Control Return and Line Feed
(b) Carriage Return and Form Feed
(c) Control Router and Line Feed
(d) Carriage Return and Line Feed

[Ans. (d) Carriage Return and Line Feed]

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

(a) py (b) xls (c) csv (d) os

[Ans. (c) csv]

4. Which of the following mode is used when dealing with non-text files like image or exe files? *[July-'22]*

(a) Text mode (b) Binary mode
(c) xls mode (d) csv mode

[Ans. (b) Binary mode]

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

(a) next() (b) skip()
(c) omit() (d) bounce()

[Ans. (a) next()]

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

(a) Line Terminator (b) Enter key
(c) Form feed (d) Data Terminator

[Ans. (a) Line Terminator]

7. What is the output of the following program?

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

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

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

(a) chennai,mylapore
(b) mumbai,andheri
(c) chennai
mumbai
(d) chennai,mylapore
mumbai,andheri

[Ans. (b) mumbai,andheri]

8. Which of the following creates an object which maps data to a dictionary? *[PTA-1]*

(a) listreader() (b) reader()
(c) tuplereader() (d) DictReader ()

[Ans. (d) DictReader ()]

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

(a) Editing (b) Appending
(c) Modification (d) Alteration

[Ans. (c) Modification]

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

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

(a) Exam Quarterly Halfyearly
(b) Exam Quarterly Halfyearly
(c) E
Q
H
(d) Exam,
Quarterly,
Halfyearly

[Ans. (d) Exam, Quarterly, Halfyearly]

Part - II

Answer the following questions (2 marks)

1. What is CSV File? *[PTA-3; Aug-2021; May-'22]*

Ans. (i) A CSV file is a human readable text file where each line has a number of fields, separated by commas or some other delimiter.

(ii) A CSV file is also known as a Flat File that can be imported to and exported from programs that store data in tables, such as **Microsoft Excel or OpenOfficeCalc**.

2. Mention the two ways to read a CSV file using Python. *[PTA-2; Sep-2020]*

Ans. There are two ways to read a CSV file.

(a) Use the csv module's reader function

(b) Use the DictReader class.

3. Mention the default modes of the File.

Ans. (i) The default is reading ('r') in text mode.

(ii) In this mode, while reading from the file the data would be in the format of **strings**.

4. What is use of next() function?

Ans. (i) **"next()" command** is used to avoid or skip the first row or row heading.

(ii) **Example :** While sorting the row heading is also get sorted, to avoid that the first is skipped using next().

(iii) Then the list is sorted and displayed.

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

Ans. To sort by more than one column you can use itemgetter with multiple indices: operator .itemgetter (1,2).

Syntax :

sortedlist = sorted(data, key=operator.itemgetter(Col_number),reverse=True)

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

PART - III

ANSWER THE FOLLOWING QUESTIONS (3 MARKS)

1. Write a note on open() function of python. What is the difference between the two methods? *[PTA-1; HY-2019; July-'22]*

Ans. Python has a built-in function open() to open a file. This function returns a **file object**, also called a **handle**, as it is used to read or modify the file accordingly.

(i) The default is reading in text mode.

(ii) In this mode, while reading from the file the data would be in the format of **strings**.

(iii) On the other hand, binary mode returns bytes and this is the mode to be used when dealing with non-text files like image or exe files.

2. Write a Python program to modify an existing file. *[July-'22]*

Ans.

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

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

Ans.

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

UNIT-V INTEGRATING PYTHON WITH MYSQL AND C++

CHAPTER 14

Importing C++ Programs in Python

CHAPTER SNAPSHOT

Evaluation

Part - I

Choose the best answer (1 mark)

1. Which of the following is not a scripting language?

(a) JavaScript (b) PHP
(c) Perl (d) HTML

[Ans. (d) HTML]

2. Importing C++ program in a Python program is called *[HY-2019]*

(a) wrapping (b) Downloading
(c) Interconnecting (d) Parsing

[Ans. (a) wrapping]

3. The expansion of API is

(a) Application Programming Interpreter
(b) Application Programming Interface
(c) Application Performing Interface
(d) Application Programming Interlink

[Ans. (b) Application Programming Interface]

4. A framework for interfacing Python and C++ is *[Mar.-2020; May-'22]*

(a) Ctypes (b) SWIG
(c) Cython (d) Boost

[Ans. (d) Boost]

5. Which of the following is a software design technique to split your code into separate parts? *[PTA-5]*

(a) Object oriented Programming
(b) Modular programming
(c) Low Level Programming
(d) Procedure oriented Programming

[Ans. (b) Modular programming]

6. The module which allows you to interface with the Windows operating system is *[Aug-2021]*

(a) OS module (b) sys module
(c) csv module (d) getopt module

[Ans. (a) OS module]

7. getopt() will return an empty array if there is no error in splitting strings to *[July-'22]*

(a) argv variable (b) opt variable
(c) args variable (d) ifile variable

[Ans. (c) args variable]

8. Identify the function call statement in the following snippet. *[Govt. MQP-2019]*

```
if __name__ =='__main__':
main(sys.argv[1:])
```

(a) main(sys.argv[1:]) (b) __name__
(c) __main__ (d) argv

[Ans. (b)__name__]

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

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

[Ans. (d) PYTHON]

10. What does __name__ contains ? *[PTA-6]*

(a) c++ filename (c) main() name
(c) python filename (d) os module name

[Ans. (C) python filename]

Part - II

Answer the following questions (2 marks)

1. What is the theoretical difference between Scripting language and other programming language? *[Aug-2021]*

Ans. (i) The theoretical difference between the two is that scripting languages do not require the compilation step and are rather interpreted.

(ii) For example, normally, a C++ program needs to be compiled before running whereas, a scripting language like JavaScript or Python need not be compiled.

(iii) A scripting language requires an interpreter while a programming language requires a compiler.

2. Differentiate compiler and interpreter.

Ans. *[Govt. MQP-2019; July-'22]*

S.No	Compiler	Interpreter
(i)	Compiler generates an Intermediate Code.	Interpreter generates Machine Code.
(ii)	Compiler reads entire program for compilation.	Interpreter reads single statement at a time for interpretation.
(iii)	Error deduction is difficult.	Error deduction is easy.
(iv)	Comparatively faster.	Slower.
(v)	**Example :** C++	**Example :** Python

3. Write the expansion of (i) SWIG (ii) MinGW *[PTA-1, 5; Mar.-2020]*

Ans. (i) SWIG – Simplified Wrapper Interface Generator – Both C and C++.

(ii) MINGW – Minimalist GNU for Windows

4. What is the use of modules?

Ans. (i) The use of modules to break down large programs into small manageable and organized files.

(ii) Modules provide reusability of code. Define our most used functions in a module and import it, instead of copying their definitions into different programs.

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

Ans. 'cd' command refers to change directory and absolute path refers to the complete path where python is installed.

(Eg) "cd:\>cd c:\program files\open office 4\program"

Part - III

Answer the following questions (3 Marks)

1. Differentiate PYTHON and C++.

Ans. *[HY-2019; Aug-2021]*

S. No	PYTHON	C++
(i)	Python is typically an "interpreted" language	C++ is typically a "compiled" language
(ii)	Python is a dynamic-typed language	C++ is compiled statically typed language
(iii)	Data type is not required while declaring variable	Data type is required while declaring variable
(iv)	It can act both as scripting and general purpose language	It is a general purpose language

2. What are the applications of scripting language? *[PTA-4; Sep-2020]*

Ans. (i) To automate certain tasks in a program

(ii) Extracting information from a data set

(iii) Less code intensive as compared to traditional programming language

(iv) can bring new functions to applications and glue complex systems together

3. What is MinGW? What is its use?

Ans. (i) MinGW refers to a set of runtime header files, used in compiling and linking the code of C, C++ and FORTRAN to be run on Windows Operating System.

(ii) MinGw-W64 (version of MinGW) is the best compiler for C++ on Windows. To compile and execute the C++ program, you need 'g++' for Windows. MinGW allows to compile and execute C++ program dynamically through Python program using g++.

(iii) Python program that contains the C++ coding can be executed through either by using command prompt or by using run terminal.

4. Identify the module, operator, definition name for the following.

welcome.display() *[PTA-6; July-'22]*

Ans.

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

5. What is sys.argv? What does it contain? *[May-'22]*

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

main(sys.argv[1]) :

(i) Accepts the program file (Python program) and the input file (C++ file) as a list(array).

(ii) argv[0] contains the Python program which is need not to be passed because by default _main_contains source code reference.

(iii) argv[1] contains the name of the C++ file which is to be processed.

Part - IV

Answer the following questions (5 marks)

1. Write any 5 features of Python. *[PTA-3; Mar-2020]*

Ans. (i) Python uses Automatic Garbage Collection.

(ii) Python is a dynamically typed language.

(iii) Python runs through an interpreter.

(iv) Python code tends to be 5 to 10 times shorter than that written in C++.

(v) In Python, there is no need to declare types explicitly.

(vi) In Python, a function may accept an argument of any type, and return multiple values without any kind of declaration beforehand.

2. Explain each word of the following command Python <filename.py> -<i> <C++ filename without cpp extension> *[May-'22]*

Ans. Python <filename.py> -i <C++ filename without cpp extension>

where,

Python	Keyword to execute the Python program from command-line
filename.py	Name of the Python program to executed
-i	input mode
C++ filename without cpp extension	Name of C++ file to be compiled and executed

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

Ans. (i) **Python's sys module :** This module provides access to built in variables used by the interpreter. One among the variable in sys module is argv.

sys.argv :

(i) sys.argv is the list of command-line arguments passed to the Python program. **argv contains** all the items that come along via the command-line input, it's basically a list holding the command-line arguments of the program.

(ii) To use sys.argv, import sys should be used. The first argument, sys.argv[0] contains the name of the python program (example pali.py) and sys.argv [1]is the next argument passed to the program (here it is the C++ file), which will be the argument passed through main ().

(ii) **Python's OS Module :**

(i) The **OS** module in Python provides a way of using operating system dependent functionality.

(ii) The functions that the **OS** module allows you to interface with the Windows operating system where Python is running on.

os.system():

(i) Execute the C++ compiling command (a string contains Unix, C command which also supports C++ command) in the shell (Here it is Command Window).

(ii) For Example to compile C++ program **g++ compiler** should be invoked.

(iv) **Command :** os.system ('g++' + <varaiable_name1> '-<mode>' + <variable_name2>

(i)	os.system:-	function system() defined in os module to interact with the operating system
(ii)	g++:-	General compiler to compile C++ program under Windows Operating system.
(iii)	variable_name1:-	Name of the C++ file along with its path and with extension .cpp in string format.
(iv)	mode:-	To specify input or output mode. Here it is o prefixed with hyphen.
(v)	variable_name2:-	Name of the executable file with extension .exe in string format.

Python getopt module :

(i) The getopt module of Python helps you to parse (split) command-line options and arguments.

(ii) This module provides getopt() method to enable command-line argument parsing.

(iii) **Python getopt.getopt method :**

(i) This method parses command-line options and parameter list. Following is the syntax for this method –

(ii) <opts>,<args>=getopt.getopt(argv, options, [long_options])

- **argv** - This is the argument list

of values to be parsed (splited). In our program the complete command will be passed as a list. **For example:**

c:\pyprg\pali.py -i c:\pyprg\pali_cpp

- **options** - This is string of option letters that the Python program recognize as, for input or for output, with options (like 'i' or 'o') that followed by a colon (:).

  Here colon is used to denote the mode.
- **long_options** - This contains a list of strings. Argument of Long options should be followed by an equal sign ('=').
- In our program the C++ file name along with its path will be passed as string and 'i' i will be also passed to indicate it as the input file.

(iv) **getopt()** method returns value consisting of two elements.

(v) Each of these values are stored separately in two different list (arrays) opts and args.

(vi) **Opts** contains list of splitted strings like mode and path. args contains error string, if at all the comment is given with wrong path or mode.

(vi) **args** will be an empty list if there is no error.

(vii) **Example :**

opts, args = getopt.getopt (argv,
"i:",['ifile='])

- where opts contains - ('i', 'c:\\pyprg\\p4')]
- -i: - option mode should be followed by : (colon)
- 'c:\\pyprg\\p4' - value absolute path of C++ file.

(viii) In our examples since the entire command line commands are parsed and no leftover argument, the second argument args will be empty [].

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

(x) **Example :**

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

4. Write the syntax for getopt() and explain its arguments and return values. *[PTA-2, 5]*

Ans. **Python getopt Module :**

(i) The getopt module of Python helps you to parse (split) command-line options and arguments.

(ii) This module provides getopt() method to enable command-line argument parsing.

getopt.getopt method : This method parses command-line options and parameter list. Following is the syntax for this method –

<opts>,<args>=getopt.getopt(argv, options, [long_options])

Here is the detail of the parameters –

(i) **argv :** This is the argument list of values to be parsed (splited). In our program the complete command will be passed as a list.

(ii) **options :** This is string of option letters that the Python program recognize as, for input or for output, with options (like 'i' or 'o') that followed by a colon (:). Here colon is used to denote the mode.

(iii) **long_options :** This contains a list of strings. Argument of Long options should be followed by an equal sign ('='). In our program the C++ file name along with its path will be passed as string and 'i' i will be also passed to indicate it as the input file.

getopt() method returns value consisting of two elements. Each of these values are stored separately in two different list (arrays) **opts** and **args**. Opts contains list of splitted strings like mode and path. args contains error string, if at all the comment is given with wrong path or mode. args will be an empty list if there is no error.

For example, The Python code which is going to execute the C++ file p4 in command line will have the getopt() method like the following one.

opts, args = getopt.getopt (argv, "i:",['ifile='])

where **opts** contains	[('-i', 'c:\\pyprg\\p4')]
-i :-	**option** nothing but **mode should be followed by :**
'c:\\pyprg\\p4'	**value** nothing but the **absolute path of C++ file.**

In our examples since the entire command line commands are parsed and no leftover argument, the **second argument args** will be empty []. If args is displayed using print() command it displays the output as [].

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

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

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

Ans. #Now select File→New in Notepad and type the Python program as main.py

```
# Program that compiles and executes a .cpp file
# Python main.py -i welcome
import sys, os, getopt
def main(argv):
    cpp_file = ''
    exe_file = ''
    opts, args = getopt.getopt(argv, "i:",['ifile='])
    for o, a in opts:
        if o in ("-i", "--ifile"):
        cpp_file = a + '.cpp'
        exe_file = a + '.exe'
        run(cpp_file, exe_file)
def run(cpp_file, exe_file):
    print("Compiling " + cpp_file)
    os.system('g++ ' + cpp_file + ' -o ' + exe_ file)
    print("Running " + exe_file)
    print("-------------------------")
    print
    os.system(exe_file)
    print
if __name__ =='__main__':
    main(sys.argv[1:])
```

Output :

```
WELCOME
```

Hands on Experience

1. Write a C++ program to create a class called Student with the following details

Protected member

Rno integer

Public members

void Readno(int); to accept roll number and assign to Rno

void Writeno(); To display Rno.

The class Test is derived Publically from the Student class contains the following details

Protected member

Mark1 float

Mark2 float

Public members

void Readmark(float, float); To accept mark1 and mark2

void Writemark(); To display the marks

Create a class called Sports with the following detail

Protected members

score integer

Public members

void Readscore(int); To accept the score

void Writescore(); To display the score

The class Result is derived Publically from Test and Sports class contains the following details

Private member

Total float

Public member

void display() assign the sum of mark1, mark2, score in total.

invokeWriteno(), Writemark() and Writescore(). Display the total also.

Save the C++ program in a file called hybrid. Write a python program to execute the hybrid.cpp

Ans. In Notepad, type the C++ program.

```
#include<iostream>
using namespace std;
class student
{
protected:
        int no;
public:
void readno(int rollno)
{
        mo = rollno;
}
void writeno()
{
        cout<<"\n Roll no."<<rno;
}};
class test: public student
{
        protected:
        float mark1,mar2;
        public:
void readmark(float m1, float m2)
{
        mark1 = m1;
        mark2 = m2;
}
void writemark()
{
        cout<<"\n mark1"<<mark1;
        cout<<"\n mark2"<<mark2;
}};
class sports
{
        protected:
            int score;
        public:
void readscore(int s)
{
        score = s;
}
void writescore()
{
        cout<<"SCORE:"<<score;
}};
class result : public test, public sports
{
        private:
        float total;
        public;
        void display()
        {
            total = mark1 + mark2;
            cout<<"TOTAL MARKS: "<<total;
}};
int main()
{
        result r;
        r.readno(5);
        r.readmark(100,100);
        r.readscore(200);
        r.writeno();
        r.writemark();
        r.display();
        r.writescore();
        return ();
}
```

save this file as hybrid.cpp

Now type the python program in New Notepad file.

```
#python hybrid.py -i hybrid.cpp
import sys,os,getopt
def main(argv);
        cpp_file="
        exe_file="
        opts, args = getopt.getopt(argv, "i:",
                                        [ifile='])
        for o, a in opts:
        if o, a in opts:
        cpp_file=a+'.cpp'
        exe_file=a+'.exe'
        run(cpp_file, exe_file)
def run(cpp_file, exe_file)
        print("Compiling"+cpp_file)
        os.system('g++'+ cpp_file + '-o'+ exe_file)
        print("Running" + exe_file)
        print("------------------")
        print
        os.system(exe_file)
        print
if__name__=='__main__':
main(sys.argv[1:])
```

Output :

```
        Rollno   : 5
        Mark1    : 100
        Mark2    : 100
        TOTAL MARKS : 200
        SCORE : 200
```

2. Write a C++ program to print boundary elements of a matrix and name the file as Border.cpp. Write a python program to execute the Border.cpp

Ans. Select File → New in Notepad and type the C++ program.

```
#include<iostream>
#include<bits/stdc++.h>
using namespace std;
        const int MAX = 100;
void printBoundary(int a[][max], int m, int n)
{
        for(int i=0; i < m; i++)
        {
            for(int j=0; j < n; j++)
            {
                if(i= =0 || j= =0 || i= =n–1 ||
                                        j= =n–1)
                    cout<<a[i][j]<<" ";
                else
                    cout <<" "
                    cout << " ";
            }
                    cout <<"\n;
            }}
int main()
{
        int  a[4][MAX]  =  {  {1,2,3,4},  {5,6,7,8},
                                {1,2,3,4}, {5,6,7,8}};
        print Boundary(a,4,4);
        return 0;
}
```

save it as Border.cpp

open a New notepad file and type the python program to execute border.cpp

```
#python border.py -i border.cpp
import sys,os,getopt
def main(argv):
        cpp_file ="
        exe_file ="
        opts, args= getopt.getopt(argv, "i:",
                                        ['ifile="])
        for o, a in opts:
        if o in("-i", "--ifile"):
        cpp_file = a+'.cpp'
        exe_file = a+'.exe'
        run(cpp_file, exe_file)
def run(cpp_file, exe_file):
        print("Compiling" + cpp_file)
        os.system('g++'+ cpp_file + '-o'+ exe_file)
        print("Running" + exe_file)
        print("---------------")
        print
        os.system(exe_file)
        print
        if__name__=='__main__':
        main(sys.argv[1:])
```

Output :

```
1   2   3   4
5           8
1           4
5   6   7   8
```

PTA QUESTIONS AND ANSWERS

1 MARK

1. **Which of the following is the special variable which by default stores the name of the file?** *[PTA-1]*

   (a) __name__ (b) __init__
   (c) __del__ (d) __def__

   [Ans. (a) __name__]

2. **_______ is a built-in variable which evaluates to the name of the current module.** *[PTA-4]*

   (a) __name__ (b) __main__
   (c) __mode__ (d) __init__

   [Ans. (a) __name__]

GOVERNMENT EXAM QUESTIONS AND ANSWERS

1 MARK

1. **Which of the following is not a scripting language?** *[Sep-2020]*

   (a) Ruby (b) DBMS
   (c) Perl (d) JavaScript

   [Ans. (b) DBMS]

2 MARK

1. **Write the syntax of getopt.getopt method.** *[May-'22]*

Ans. <opts>,<args> = getopt.getopt(argv, options, [long_options]).

ADDITIONAL QUESTIONS AND ANSWERS

CHOOSE THE CORRECT ANSWER 1 MARK

1. **Which of the following are general purpose programming language?**

   (a) Python (b) C++
   (c) Java (d) All of these

   [Ans. (d) All of these]

2. **Which of the following is not general purpose language?**

   (a) Python (b) Perl
   (c) Java (d) C++

   [Ans. (b) Perl]

19. The command to change to the folder where Python is located is
(a) Change (b) CD
(c) Dir (d) CDir
[Ans. (b) CD]

20. The syntax to execute the python program is
(a) Python –i <filename.Py> <C++ filename>
(b) Python <filename – py> <C++ filename> -i
(c) Python <C++ filename> -i <filename.py>
(d) Python <filename.py> -i <C++ filename>
[Ans. (d) Python <filename.py> -i <C++ filename>]

21. In the command python <filename.py> - i <C++ filename> where i denotes.
(a) Information (b) Input mode
(c) ios (d) Interpreter
[Ans. (b) Input mode]

22. Which of the following is not a python module?
(a) OS (b) Sys
(c) Tel (d) Getopt
[Ans. (c) Tel]

23. The operator used to access the python functions using modules is
(a) . (b) : (c) , (d) : :
[Ans. (a) .]

24. Which of the following is not a python module?
(a) Sys (b) OS
(c) Getopt (d) g++
[Ans. (d) g++]

25. Which of the following is a python module?
(a) Sys (b) OS
(c) Getopt (d) All of these
[Ans. (d) All of these]

26. Which of the following is an array holding the command line arguments of the program?
(a) g++ (b) argv
(c) Opts (d) Getopt
[Ans. (b) argv]

27. How many options getopt provides to enable command line argument parsing?
(a) 3 (b) 7 (c) 2 (d) 4
[Ans. (c) 2]

28. Getopt () method returns values are started in
(a) Opts (b) Args
(c) Sys (d) a and b
[Ans. (d) a and b]

29. The mode 'i'/'o' parses each values of the command line and pass as argument to the list called
(a) Args (b) Opts
(c) Sys (d) Argv
[Ans. (b) Opts]

30. Which of the following definition invoke the 'g++' compiler and creates the exe file?
(a) Main (b) Name
(c) Run (d) System
[Ans. (c) Run]

Match the following

1.

1	PHP	Interface with C
2	HTML	Interface generator both C & C++
3	SWIG	Programming language
4	API	Scripting language

(a) 1-2-3-4 (b) 4-3-2-1
(c) 4-2-3-1 (d) 4-1-2-3
[Ans. (b) 4-3-2-1]

Choose the odd man out

1. (a) Java (b) Python
(c) HTML (d) C++[Ans. (b) Python]

Choose and fill in the blanks

1. ______ is typically interpreted language.
(a) Python (b) Java
(c) C++ (d) None of these
[Ans. (a) Python]

2. ______ is mostly used as a 'glue' language.
(a) C++ (b) Java
(c) Python (d) CSV
[Ans. (c) Python]

12th STD

INSTANT SUPPLEMENTARY EXAM - JULY 2022

Reg. No. □□□□□□□

COMPUTER SCIENCE (with Answers)

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. **(15×1=15)**

(ii) Choose the most appropriate answer from the given **four** alternatives and write the option code and the corresponding answer.

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

(a) Subroutines (b) Function
(c) Files (d) Modules

2. Which of the following functions build the abstract data type?

(a) Constructors (b) Destructors
(c) Recursive (d) Nested

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

(a) Local Scope (b) Global Scope
(c) Module Scope (d) Function Scope

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

(a) Ctrl + C (b) Ctrl + F
(c) Ctrl + B (d) Ctrl + N

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

(a) Semicolon (;) (b) Dollar ($)
(c) Comma (,) (d) Colon (:)

6. Which is the most Comfortable loop?

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

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

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

8. What is Stride?

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

9. What will be result of the following Python code

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

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

10. Class members are accessed through which operator?

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

11. A table is known as :

(a) Tuple (b) Attribute
(c) Relation (d) Entity

12. The command to delete a table is :

(a) DROP (b) DELETE
(c) DELETE ALL (d) ALTER TABLE

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

(a) Text mode (b) Binary mode
(c) xls mode (d) CSV mode

14. getopt () will return an empty array if there is no error in splitting strings to :

(a) argv variable (b) opt variable
(c) args variable (d) ifile variable

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

(a) MAX () (b) LARGE ()
(c) HIGH () (d) MAXIMUM ()

PART - II

Note : Answer **any six** questions. Question No. **24** is **compulsory**. **6 × 2 = 12**

16. Differentiate constructor and selector.

17. Write a short note on Namespaces.

18. Write note on range ( ) in loop.

19. What is meant by scope of variables? Mention its types.

20. What is set in Python?

21. Mention few examples of Database Management System.

22. Differentiate compiler and interpreter.

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

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

```
Str1 = "School"
print (str1*3)
```

PART - III

Note : Answer **any six** questions. Question No. **33** is **compulsory.** **6 × 3 = 18**

25. Write the syntax of 'while' loop.

26. Write the basic rules for global keyword in python.

27. Identify the module, operator, definition name for the following:

welcome.display()

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

29. Write a python program to modify an existing file.

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

31. Write a note on open ( ) function of Python. What are the difference between its two methods?

32. List the general types of data visualization.

33. What is the output of the following program?

```
class Greeting :
        def _ init _ (selt, name) :
                self._ name = name
        def display (self) :
                print ("Good Morning", self._ name)
obj = Greeting ('Bindu Madhavan")
obj.display ( )
```

PART - IV

Note : Answer **all** the **questions:** **5 × 5 = 25**

34. (a) Discuss about linear search algorithm.

(OR)

(b) Explain the different types of function with an example.

35. (a) What is nested tuple? Explain with an example.

(OR)

(b) Explain about SQLite and the steps to be used.

36. (a) Write a detail note on 'for' loop.

(OR)

(b) Explain the different types of operators used in Python:

37. (a) Write the different types of constraints and their functions?

(OR)

(b) Explain the characteristics of DBMS.

38. (a) Write the different methods to read a file in Python.

(OR)

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

❁❁❁❁❁

ANSWER

PART - I

1. (b) Function

2. (a) Constructors

3. (a) Local Scope

4. (d) Ctrl + N

5. (c) Comma (,)

6. (c) for

7. (d) def

8. (d) third argument of slice operation

9. (b) [0, 1, 4, 9, 16]

10. (b) .

11. (c) Relation

12. (a) DROP

13. (b) Binary mode

14. (c) args variable

15. (a) MAX ( )

PART - II

16.

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

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

Example : a : = 5

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