

①

52717

SET 3

Q. 1. (2 marks each)

Q.	Soln	Q.	Soln	vii	b
i	a	iv	a	viii	a
ii	b	v	b	ix	a
iii	b	vi	c	x	d

Q 2 a) . Attempt any ONE question (8 Marks).

i) Methodology of Problem Solving

The process of developing a solution consists of development of a structure chart, a pseudo code and a flow chart.

(1) Structure chart: A structure chart is a hierarchy that shows the functional flow of a program. Large programs are complex structures comprising interrelated parts. Hence, they must be very carefully laid out. A structure chart shows a logical break-down of a program into different steps. Each step has separate modules that are related to different modules. Before writing any program, it is important to design a structure in line with the structure chart.







(2) Pseudo code A pseudo code is used to state an algorithm in an English-like syntax.

A pseudo code is used by almost all the professional programmers. Pseudo codes are easy to understand. There are used to depict the design of an algorithm.

(3) Flow chart A flowchart is a simple diagram that illustrates a sequence of operations to be performed for obtaining a solution. A flow chart is a graphical representation of the logical flow of data. It uses the standard graphical symbols to narrate the sequential process of a specific module. These steps must be followed while designing the whole program

ii) Flowchart: (2+6 M)

A flowchart is a simple diagram that illustrates a sequence of operations to be performed for obtaining a solution. There are six basic symbols that are used to draw a simple flow chart which are as follows:

Symbol	Operation	Meaning
	Start-stop	Represents the beginning and the end of the flow chart
	Input/output	Represents the values to be given by the end user and the result to be displayed
	Processing	Represents the arithmetic operations to compute a value.
	Decision making/Checking	Represents the logical checking to decide the flow of the sequence
	Connection	Represents the continuing of flowchart in another place of page
	Flow lines	Represents the direction of flow

Q2 b) Attempt any TWO questions. (12 M)

i) three dimensions basis on which a problem is analyzed: Decomposable/Non-decomposable:

Analyse whether a problem is decomposable, i.e., whether it can be broken down into sub-problems. If a problem cannot be decomposed, then such a problem is non-decomposable.

(2) Solution steps-Can/Cannot be ignored: It is important to determine if we can/cannot ignore a few steps while solving a problem. On the basis of this key dimension, a problem can be categorised into the following:

- Ignorable: In case of ignorable problems, a few steps can be ignored since they are not extremely important to follow for solving the problem.
- Recoverable: Recoverable problems are those in which previously executed steps can be easily backtracked.
- Irrecoverable: In case of irrecoverable problems, we cannot backtrack

Previously executed steps; hence it is important to carefully solve the problem.

(3) Predictable/Unpredictable: This dimension determines the certain/uncertainty factor attached to a problem it is important to know whether there will be an expected/unexpected output after applying a particular input.

2

SET 3

(4) Good solution: Absolute/relative: It is important to determine whether we are trying to achieve an optimum solution or we want to achieve the best solution. On the basis of this decision, appropriate techniques can be applied to achieve the best solution.

ii) Statement and Expression in Python: 6+6

A statement is an instruction that can be interpreted by the Python interpreter.

A statement is interpreted by the interpreter and after execution displays some result.

A program is formed by a sequence of one or more statements. A statement will have internal components (e.g. expressions). Statements are everything that can make up a line (or several lines) of Python code. Note that expressions are statements as well. Examples:
Each of the following line is a statement in Python.

```
>>>print (42)
```

```
>>>if x = 10:
```

```
    y = 5
```

```
    a = 7
```

(II) An expression is a combination of variables, operators, values and reserved Keywords.

Whenever we type an expression in the command line, the interpreter evaluates it and produces the result.

The following are valid expressions:

```
(1) >>>42
```

```
Output: 42
```

```
(2) >>> n = 17
```

```
Output: 17
```

```
(3) >>> n = 17
```

```
>>> n + 25
```

```
Output: 42
```

iii) What are data types in Python?(2+2+2+2 with example for each data type)

a) a data type consists of a set of values and a set of operations that can be performed on those values.

Data types specify the type of values that can be stored.

Python has six basic data types as follows:

(1) Numeric (2) String (3) List (4) Tuple

(5) Dictionary (6) Boolean

b) Numeric: Numeric data can be broadly divided into integers and real numbers

c) String: Single quotes or double quotes are used to represent strings. A string Python is a sequence of alphabets, numbers and special characters.

d) Boolean: Boolean type is used when we want to test a particular condition during the execution of the program. There are only two values that a Boolean type can take: true or false. Boolean type is denoted by the keyword bool. All comparison operators return boolean type values. Boolean values are often used in selection and iteration statements.

iv) Concatenation: (2+4 M)

We can join two or more strings to form a new string using the concatenation operator +.

Example

```
>>>print("Hi" + " " + "there" + " " + "John")
```

Output: Hi there John

Q 3 a) Attempt any ONE question : (8 Marks).

i) Looping statements in Python:(2+3+3 M)

We can create loops by using the iteration statements. Repeated execution of a set of statements is called iteration. Iteration statements causes embedded statements to be executed a number of times, subject to loop-termination criteria.

a) for: The for loop in python is used to iterate over a sequence or other iterable objects.

Syntax: for val in sequence:

statement

b) while: For repeating Python statements while a condition is true. A while loop executes an unknown number of times, as long as the expression is true.

Syntax: while expression:

Statement

ii) Recursion in python:(2+4 M)

The recursion is a way of programming or coding a problem, in which a function calls itself one or more times in its body. Usually, it is returning the return value of this function call. Python stop calling recursive function after 1000 calls by defaults. **+Program**

```
>>>def fib(n):
```

```
    if n<= 1:
```

```
        return n
```



SET 3

```
else:
    return(fib(n-1)+fib(n-2))
nt = 4
for I in range (nt):
    print(fib(i))
output : 0 1 1 2
```

Q3 b) Attempt any TWO questions. (12 Marks)

i) function within the function.(6 M)

ii) for statement:

The for loop in python is used to iterate over a sequence or other iterable objects.

Syntax: for val in sequence:

```
    statement
>>>for i in range (2,12,2):
    print i
```

iii) Adding new functions:

Step 1: Use def keyword to declare the function followed by the name of function.

Step 2: Write the arguments inside the parentheses of the function, and end with a colon.

Step 3: Add the program statements to be executed.

Step 4:End the function with / without statement.

iv)Maximum number:

```
>>>def FnMax(s,t):
    if s > t:
        print(s,"is maximum")
    elif s == t:
        print(s, "is equal to ", t)
    else:
        print(t,"is maximum")
```

Q 4 a) Attempt any ONE question : (8 Marks).

i) String in Python:

A string in Python can be a series or a sequence of alphabets, numbers or special character. String are created by enclosing various character within single quotes or double quotes.

(I) >>> S=" elephantiasis"

(II) >>> print(S[3:])
phantiasis # o/p

(III) >>>print(S[0:8])
elephant #o/p

ii)Range function in a list: (2+2+2+2 M)

Range function: The built-in range function is very useful to generate sequence of numbers in the form of a list. The given end point is never the part of the generated list.

```
>>> L = [25, 30, 37, 89, 49]
```

```
(I)>>>for i in range(5):
```

```
    print L[i]
```

```
o/p: 25 30 37 89 49
```

```
(II)>>>for i in range(2,4):
```

```
    print L[i]
```

```
o/p: 37 89
```

```
(III)>>>for i in range(1,4,2):
```

```
    print L[i]
```

```
o/p: 30 89
```

Q 4 b) Attempt any TWO questions. (12 Marks).

i) (2M + 2M + 2M)

```
(I) >>>L=[3 , 2 , 'a' , 8.8 , 'abc' , [1, 2, 3] ,  
'pqr' , 150, 25 ,13]
```

```
(II) >>>L.pop(4) or del L[4]
```

```
>>>print L
```

```
[3 , 2 , 'a' , 8.8 , [1, 2, 3] , 'pqr' , 150, 25 ,13]
```

```
(III) >>>L.remove(150)
```

```
[3 , 2 , 'a' , 8.8 , [1, 2, 3] , 'pqr' , 25 ,13]
```

ii) Search in Python:

```
>>> def search(name, letter):
```

```
    for x in name:
```

```
        if x==letter:
```

```
            return(yes)
```

```
        else:
```

```
            return(no)
```

```
>>>search("Mumbai", s)
```

```
no #o/p
```

iii) >>>print S1*2

```
o/p: I respect all my teachers.I respect all my teachers.
```

```
>>>print S1+S2
```

```
o/p: I respect all my teachers.I wish to get good marks in the exam.
```

```
>>>print S2+ "I work hard for that"
```

```
o/p: I wish to get good marks in the exam.I work hard for that.
```

iv) (I)>>>D={10: 'A' , 20: 'B' , 30: 'C' , 40: 'D'}

```
>>>print D
```

```
o/p {10: 'A' , 20: 'B' , 30: 'C' , 40: 'D'}
```

4

SET 3

```
(II) >>>D[30]='z'
o/p {10: 'A', 20: 'B', 30: 'z', 40: 'D'}
(III) >>>D[50]='T'
o/p {10: 'A', 20: 'B', 30: 'z', 40: 'D', 50: 'T'}
```

Q 5 Attempt any FOUR questions. (12 Marks).

a) Logical operators:

Python has three logical operators and, or and not. The logical operators and or are used when we want to form compound conditions by combining two or more relations. An example is $a > b$ and $x == 10$.

The logical expression given above is True only if both $a > b$ and $x == 10$ are true.

If either (or both) of them are false the expression is False.

$n\%2 == 0$ or $n\%3 == 0$ is true if either or both of the conditions is (are) true, that is, if the number is divisible by 2 or 3.

The not operator negates a Boolean expression, so not $(x > y)$ is true if $x > y$ is false, that is, if x is less than or equal to y .

The operands of the logical operators should be Boolean expressions, but Python is not very strict. Any nonzero number is interpreted as True.

b) Relational Operators in Python.: (5 M)

$<$, $<=$, $>$, $>=$, $!=$, $==$

$<$, $>$, have higher precedence than $==$, $!=$, $<>$

c) Composition in python:

Function composition is a way of combining functions such that the result of each function is passed as the argument of the next function. Syntax: `FunctionB(FunctionA(x))`

In this x is the input for FunctionA and the result of that is the input for FunctionB.

d) Conditional statements:

The conditional statement causes the program control to be transferred to a specific flow based upon whether a certain condition is true or not. These are also called as selection statements.

Example: if statement, if-else statement, nested if-else

e) Traversing of Strings:

Traversal is a process in which we access all the elements of the string one by one using

some conditional statements such as for loop, while loop etc.

```
>>>var="jack john"
>>>i=0
>>>while i < len(var):
    x=var[i]
    print x
    i=i+1
```

f) pop operator: If we know the index of the element that we want to delete, then we can use the pop operator.

```
e.g.
>>>L=[10,11,12]
>>>a=L.pop(2)
>>>print L
o/p: [10,11]
>>>print a
o/p: 12
```

del operator: The del operator deletes the value on the provided index, but it does not store the value for further use.

```
e.g
>>>L=[10,11,12,14,15]
>>>del L[3]
>>>print L
o/p: [10,11,12,15]
```