

QUESTION BANK

CM5461 – PROGRAMMING WITH PYTHON

1. Define Algorithm

Algorithm: It is a sequence of instructions designed in such a way that if the instructions are executed in the specified sequence, the desired results will be obtained. The instructions in an algorithm should not be repeated infinitely. The algorithm should be written in sequence.

2. What are the properties of algorithm?

- It is written in simple English.
- Each step of an algorithm is unique and should be self
- Explanatory. An algorithm must have at least one input.
- An algorithm must have at least one
- Output. An algorithm has finite number of steps.

3. What are the building block of algorithm?

The three building block of algorithm are:

- Sequence
- Selection
- Iteration

4. What is meant by selection, iteration and sequence control structures?

Sequence:

A sequence is one of the basic logic structures in computer programming. In a *sequence* structure, an action, or event, leads to the next ordered action in a predetermined order.

Selection:

A selection (also called a decision) is also one of the basic logic structures in computer programming. In a *selection* structure, a question is asked, and depending on the answer, the program takes one of two courses of action, after which the program moves on to the next event.

Selection: A selection (also called a decision) is also one of the basic logic structures in computer programming. In a *selection* structure, a question is asked, and depending on the answer, the program takes one of two courses of action, after which the program moves on to the next event.

5. Define Flowchart

It is a pictorial representation of an algorithm. The flowchart uses different shape symbols to denote the different appropriate instructions and these instructions can be written within the boxes using clear statements.

6. Write the Characteristics of Pseudo code.

Named variables represent data and identifiers denote higher level functions.

Composed of a sequence of statements or steps.

- Statements are often numbered sequentially.
- Operational (Imperative) statements include assignment, input, and output
- Control structures provide iterative and conditional execution.
- Indentations used for grouping

8. What is need for flowchart symbol?

The need for flowchart symbols because each symbols of different shapes denotes different types of instructions. The program logic through flowcharts is made easier through the use of symbol that has standardized planning.

9. Write some rules for drawing a flowchart.

10. What are the advantages of using a flowchart

- Communication
- Effective
- Proper
- Documentation
- Efficient Coding
- Proper
- Debugging
- Efficient Program Maintenance

11. Write any two disadvantages of flowchart.

- It is not visual
- We do not get a picture of the design.

12. What is Pseudo code?

—Pseudol means imitation of false and —code| refers to the instruction written in a programming language. Pseudo code is programming analysis tool that is used for planning program logic.

13. Write the Characteristics of Pseudocode.

- Named variables represent data and identifiers denote higher level
- Functions.
- Composed of a sequence of statements or steps.
- Statements are often numbered sequentially.
- Operational (Imperative) statements include assignment, input, and
- Output.
- Control structures provide iterative and conditional execution.
- Indentations used for grouping blocks of statement

14. What are the rules for writing pseudo code?

- Write one statement
- per line. Capitalize
- initial keywords.
- Indent to show hierarchy.
- End multiline structure.
- Keep statements language independent.

15. Write any two advantages of pseudo code.

It can be done easily in any word processor.

- It can be written easily.
- It can be easily modified as compared to flowchart.

16. Write any two disadvantages of flowchart?

- It is not visual
- We do not get a picture of the design.

17. Differentiate Algorithm and Pseudo code

S.No	Algorithm	Pseudo code
1.	It is a well-defined procedure that allows a computer to solve a problem.	It is a simple way of writing programming code in English.
2.	Algorithms can be described in various ways, from pure mathematical formulas to complex graphs	Pseudo code describes how you would implement an algorithm without getting into syntactical details.

18. What is the difference between algorithm and flowchart

Sr.No	Algorithm	Flow Chart
1.	Step by Step formation	Box by box formation
2.	Mostly run to complete or manage a Program implementation.	Can be used in general terms to solve any Problem related to an individual or a group.
3.	More detailed but difficult to understand	Less detailed but easy to understand By everyone.

PART-B

1. Define algorithm. Explain in detail about the building blocks of algorithm.
2. What is flowchart? Explain the Basic design structures in Flowchart
3. What is pseudo code? Explain its guidelines and benefits.
4. Explain the design structures in pseudo code.
5. Explain the steps involved in program development cycle.
6. Write the algorithm, pseudocode and draw the flowchart for the following:
 - a. Find minimum in a list.
 - b. Insert a card in a list of sorted cards.
 - c. Guess an integer number in a range.
 - d. Towers of Hanoi.
7. Write the algorithm, pseudocode and draw the flowchart for the following:
 - a) To find the sum of square root of any three numbers.
 - b) To find the sum of first 100 integers.
 - c) To find the sum of all odd numbers till 100.
 - d) To find the sum of any five integers.
 - e) To find the factorial of number n.
 - f) To find the first n numbers in a Fibonacci series.
 - g) To find the sum of digits of a number.
 - h) To find whether a number is prime or not.
 - i) To convert temperature from Fahrenheit to Celsius
 - j) To solve the quadratic equation.
 - k) To find sum first 100 natural numbers.
 - l) To find factorial of a number.

UNIT-II

Types, Operators and Expression

1. Define python

Python is an object-oriented, high level language, interpreted, dynamic and multipurpose programming language.

2. Give the features of python.

- Easy to Use:
- Expressive Language
- Interpreted Language
- Cross-platform language
- Free and Open Source
- Object-Oriented language
- Extensible

3. What is python interpreter?

The engine that translates and runs Python is called the Python Interpreter: There are two ways to use it: immediate mode and script mode. The >>> is called the Python prompt. The interpreter uses the prompt to indicate that it is ready for instructions.

4. What is the difference between intermediate mode and script mode?

- In immediate mode, you type Python expressions into the Python Interpreter window, and the interpreter immediately shows the result.
- Alternatively, you can write a program in a file and use the interpreter to execute the contents of the file. Such a file is called a script. Scripts have the advantage that they can be saved to disk, printed, and so on.

4. What is meant by value in python?

A value is one of the fundamental things—like a letter or a number—that a program manipulates.

5. List the standard data types in python.

Python has five standard data Types:

- Numbers
- Strings
- List,
- Tuples
- Dictionary

Number data types store numeric values. Number objects are created when you assign a value to them.

Python supports four different numerical types:

- int (signed integers)
- long (long integers, they can also be represented in octal and hexadecimal)
- float (floating point real values)
- complex (complex numbers)

6. What are python strings?

Strings in Python are identified as a contiguous set of characters represented in the quotation marks. Python allows for either pairs of single or double quotes. Subsets of strings can be taken using the slice operator ([] and [:]) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.

The plus (+) sign is the string concatenation operator and the asterisk (*) is the repetition operator.

```
str = 'Hello World!'
```

```
print str[0]    # Prints first character of
the string o/p:
```

H

7. Mention the features of lists in python

Lists are the most versatile of Python's compound data types. A list contains items separated by commas and enclosed within square brackets ([]). To some extent, lists are similar to arrays in C. One difference between them is that all the items belonging to a list can be of different data type.

The values stored in a list can be accessed using the slice operator ([] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1. The plus (+) sign is the list concatenation operator, and the asterisk (*) is the repetition operator. For example

—

```
list = [ 'abcd', 786 , 2.23,'john', 70.2 ] print list[0]
```

o/p

a

b

c

d

8. What is tuple ? What is the difference between list and tuple?

- A tuple is another sequence data type that is similar to the list. A tuple consists of a number of values separated by commas.
- The main differences between lists and tuples are: Lists are enclosed in brackets ([]) and their elements and size can be changed, while tuples are enclosed in parentheses (()) and cannot be updated. Tuples can be thought of as **read-only** lists. Eg:

```
tuple = ( 'abcd', 786 , 2.23, 'john', 70.2 )
```

9. Give the features of python dictionaries

Python's dictionaries are kind of hash table type. They work like associative arrays and consist of key-value pairs. A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object. Dictionaries are enclosed by curly braces ({ }) and values can be assigned and accessed using square braces ([]). For example – dict = { }

```
dict['one'] = "This is one"
```

10. What is a variable?

One of the most powerful features of a programming language is the ability to manipulate variables. A variable is a name that refers to a value. The assignment statement gives a value to a variable.

Eg:

```
>>> n = 17
```

```
>>> pi = 3.14159
```

11. What are the rules for naming a variable?

Variable names can be arbitrarily long. They can contain both letters and digits, but they have to begin with a letter or an underscore. Although it is legal to use uppercase letters, by convention we don't. If you do, remember that case matters. Bruce and bruce are different variables.

The underscore character (_) can appear in a name.

Eg:

```
my_name
```

12. What are keywords?

Keywords are the reserved words in Python. We cannot use a keyword as variable name, function name or any other identifier. They are used to define the syntax and structure of the Python language. In Python, keywords are case sensitive. There are 33 keywords in Python.

Eg:

```
False, class, finally, return
```


13. What are the rules for writing an identifier?

- Identifiers can be a combination of letters in lowercase (a to z) or uppercase (A to Z) or digits (0 to 9) or an underscore (_). Names like myClass, var_1 and print_this_to_screen, all are valid example.
- An identifier cannot start with a digit. 1variable is invalid, but variable1 is perfectly fine. Keywords cannot be used as identifiers.
- We cannot use special symbols like !, @, #, \$, % etc. in our identifier. Identifier can be of any length.

14. what are expressions?

An expression is a combination of values, variables, operators, and calls to functions. If you type an expression at the Python prompt, the interpreter evaluates it and displays the result:

```
>>> 1 + 1=2
```

15. What is a statement?

A statement is an instruction that the Python interpreter can execute. When you type a statement on the command line, Python executes it. Statements don't produce any result. For example, a = 1 is an assignment statement. if statement, for statement, while statement etc. are other kinds of statements.

16. What is multiline statement?

In Python, end of a statement is marked by a newline character. But we can make a statement extend over multiple lines with the line continuation character (\).

For example:

```
a = 1 + 2 + 3 + \
4 + 5 + 6 + \
7 + 8 + 9
```

17. What is docstring?

Doc string is short for documentation string. It is a string that occurs as the first statement in a module, function, class, or method definition. It is used to explain in brief, what a function does.

18. What is a function? Mention the type of function and use.

A Function can be called as a section of a program that is written once and can be executed whenever required in the program, thus making code reusability.

There are two types of Functions.

- a) Built-in Functions: Functions that are predefined. We have used many predefined functions in Python.
- b) User- Defined: Functions that are created according to the requirements.

19. Mention the types of arguments in python

- 1.python default arguments.
- 2.python keyword argument
- 3.python arbitrary argument

20. What is meant by module in python?

A module is a file consisting of Python code. A module can define functions, classes and variables. A module can also include runnable code.

21. List some built in modules in python

There are many built in modules in Python. Some of them are as follows:math, random , threading , collections , os , mailbox , string , time , tkinter etc..

22. What is the use of dir() function?

The dir() built-in function returns a sorted list of strings containing the names defined by a module. The list contains the names of all the modules, variables and functions that are defined in a module.

26.What operators does python support?

- Arithmetic Operators
- Comparison (Relational) Operators
- Assignment Operator
- Logical Operators
- Bitwise Operators
- Membership Operators
- Identity Operator

27.What is an Arithmetic operator?

Arithmetic operators are used to perform mathematical operations like addition, subtraction, multiplication etc.

The operators are: +,-,/,%,*,**

28.What is the use of comparison operator?

Comparison operators are used to compare values. It either returns True or False according to the condition.

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

29. What are logical operators and Bitwise operators?

Logical operators are the and, or, not operators. Bitwise operators act on operands as if they were string of binary digits. It operates bit by bit, hence the name. The operators are:&|,^,>>,<<

30. What are assignment statements?

Assignment operators are used in Python to assign values to variables.a = 5 is a simple assignment operator that assigns the value 5 on the right to the variable a on the left.

31. Mention the features of identity operators?

is and is not are the identity operators in Python. They are used to check if two values (or variables) are located on the same part of the memory. Two variables that are equal does not imply that they are identical.

32. Give the characteristics of membership operator?

in and not in are the membership operators in Python. They are used to test whether a value or variable is found in a sequence (string, list, tuple, set and dictionary). In a dictionary we can only test for presence of key, not the value.

Operator	Meaning	Example
in	True if value/variable is found in the sequence	5 in x
not in	True if value/variable is not found in the sequence	5 not in x

PART B

1. Explain the data types in python
2. Illustrate interpreter and interactive mode in python with example.
3. Explain function and module with suitable example
4. Write short notes on types of operators in python with appropriate example
5. Explain briefly constant, variables, expression, keywords and statements available in python
6. Write the following python programs.
 - a. Exchange the value of two variables
 - b. Calculate the value of n
 - c. Test whether a given year is leap year or not
 - d. To find the sum of n natural numbers
 - e. To find whether a given number is Armstrong number or not
 - f. To print Fibonacci series
 - g. To find factorial of a given number
 - h. To convert Celsius to Fahrenheit

UNIT II

1. What is Boolean value?

A Boolean value is either true or false. In Python, the two Boolean values are True and False (the capitalization must be exactly as shown), and the Python type is bool.

```
>>> type(True)
```

2. What is Boolean Expression?

- A Boolean expression is an expression that evaluates to produce a result which is a Boolean value. For example, the operator == tests if two values are equal. It produces (or yields) a Boolean value:
- ```
>>> 5 == (3 + 2) # Is five equal 5 to the result of 3 + 2?
```
- True

### 3. What is meant by conditional If?

The if statement in python is used to test a condition. If condition is true, statement of if block is executed otherwise it is skipped. **Syntax of python if statement:**

```
if(condition):
 statements
```

### 4. What is chained conditional statement?

To check for multiple conditions elif Statement is used. This statement is like executing a if statement inside a else statement. **Syntax:** If statement:

```
 statements
elif
statement:
 statements
else:
 statements
```

### 5. Write the syntax and usage of for loop

For Loop is used to iterate a variable over a sequence (i.e., list or string) in the order that they appear. **Syntax:**

```
for <variable> in <sequence>:
```

### 6. Write the syntax and usage of while loop

While Loop is used to execute number of statements or body till the condition passed in while is true. Once the condition is false, the control will come out of the loop.

**Syntax:**

```
while
 <expression>:
 Body
```

**7.What is python break statement?**

Break statement is a jump statement that is used to pass the control to the end of the loop. Applying break statement makes the loop to terminate and controls goes to next line pointing after loop body.

**8.What is python continue statement?**

Continue Statement is a jump statement that is used to skip the present iteration and forces next iteration of loop to take place. It can be used in while as well as for loop statements.

**9.What is python pass statement?**

When you do not want any code to execute, pass Statement is used. It is same as the name refers to. It just makes the control to pass by without executing any code.

**Syntax:**

```
pass
```

**10.What is len function and explain how it is used on strings with an example.**

The len function, when applied to a string, returns the number or character in a string.

Example:

```
>>>book='Problem Solving and Python Programming'
>>>l
en(book)
38
>>>
```

**11.Explain about string slicing with examples.**

A substring of a string is obtained by taking a slice. The operator [n:m] returns the part of the string from the nth character to the mth character, including the first but excluding the last.

```
>>>book='Problem Solving and Python
Programming' >>>print(book[0:7])
Problem
>>>print(book[21:27])
python
```

## 12. What are the two operators that are used in string functions?

The in operator tests for membership.

```
>>>'V' in _VRB'
```

```
True
```

```
>>>'S' in _VRB'
```

```
>>>False
```

The not in operator returns the logical opposite results of in operator.

```
>>>'x' not
```

```
in _VRB'
```

```
True
```

## 13. What is the use of str.upper() and str.lower() functions in string?

The functions str.upper() and str.lower() will return a string with all the letters of original string converted to upper or lower case letters.

```
>>>ss='VRB Publishers'
```

```
>>>print(ss.
```

```
upper())
```

```
VRB
```

```
PUBLISHE
```

```
RS
```

```
>>>print(ss
```

```
.lower())
```

```
Vrb
```

```
publishers
```

## 14. Explain string comparison with an example.

The comparison operator works on string to check if two strings are equal.

```
>>>word='VRB Publishers'
```

```
>>>if word=='VRB
```

```
Publishers' Print(_Both
```

```
are Equal')
```

```
Both are Equal
```

## 15. How to split strings and what function is used to perform that operation?

The str.split() method is used to split strings up.

```
>>>book='Problem Solving and Python Programming'
```

```
>>>print(book.split())
```

```
[_Problem', _Solving', _and', _Python', _Programing']
```

## **PART B**

1. Explain the function arguments in python
2. Explain call by value and call by reference in python
3. Briefly explain about function prototypes
4. Answer the following questions.

What is String? How do u create a string in Python? (4 marks)

How to perform a user input in Python? Explain with example. (4 marks)

Write a program to check whether entered string is palindrome or not.

5. Answer the following questions.

What is the output produced by this program given below?

(8 marks) words = \_Hello

World' print words.title()

print

words.replace(\_Worldl,'Osw

al') print words.upper()

print words\*2

Assuming num=125, determine the value of each of the following

Python expressions. (8 marks)

(i) num/125

(ii) num% 100

(iii) (num==2 1)&(2<3)

(iv) not((num<45.9)&(6\*2<=13))



## UNIT- III

### Data Structure

#### 1. What are tuples in Python?

A tuple is another sequence data type that is similar to the list. A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses.

#### 2. What is the difference between tuples and lists in Python?

The main differences between lists and tuples are –

Lists are enclosed in brackets ( [ ] ) and their elements and size can be changed, while tuples are enclosed in parentheses ( ( ) ) and cannot be updated. Tuples can be thought of as read- only lists.

#### 3. What are Python's dictionaries?

Python's dictionaries are kind of hash table type. They work like associative arrays or hashes found in Perl and consist of key-value pairs. A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object.

#### 4. Explain how to create a dictionary in python?

Dictionaries are enclosed by curly braces ( { } ) and values can be assigned and accessed using square braces ( [ ] ).

```
dict = {}
dict['one'] = "This is one"
dict[2] = "This is two"
tinydict = {'name': 'john', 'code':6734, 'dept': 'sales'}
```

#### 5. Explain what is range() function and how it is used in lists?

The range function returns an immutable sequence object of integers between the given start integer to the stop integer

range(start,stop,[step])

```
>>>f
```

```
or l
```

```
in
```

```
range(1,10,2):
```

```
print(i,end=" ") 1 3 5 7 9
```

## 6. How lists are updated in Python?

The append() method is used to add elements to a list.

Syntax:

```
list.append(obj)
```

```
List=[123,'VR
```

```
B']
```

```
List.append(2017)
```

```
Print(—
```

```
Updat
```

```
ed
```

```
List:l,
```

```
List)
```

```
Output: Updated List: [123,'VRB',2017]
```

## 7. Write a few methods that are used in Python Lists.

- append()- add an element to end of list
- insert()- insert an item at the defined index
- remove()- removes an item from the list
- clear()- removes all items from the list
- reverse()- reverse the order of items in the list

## 8. What are the advantages of Tuple over List?

- Tuple is used for heterogeneous data types and list is used for homogeneous data types.
- Since tuple are immutable, iterating through tuple is faster than with list.
- Tuples that contain immutable elements can be used as key for dictionary.
- Implementing data that doesn't change as a tuple remains write-protected.

## 9. What is indexing and negative indexing in Tuple?

The index operator is used to access an item in a tuple where index starts from 0.

Python also allows negative indexing where the index of -1 refers to the last item, -2 to the second last item and so on.

```
>>>my_tuple=('p','y','t','h','o','n')
```

```
>>>print(my_tuple[5]
```

```
) n
```

```
>>>print(m
```

```
y_tuple[-
```

```
6]) p
```

## 10. What is the output of print tuple[1:3] if tuple = ( 'abcd', 786 , 2.23, 'john', 70.2 )? In

the given command, tuple[1:3] is accessing the items in tuple using indexing.

It will print elements starting from 2nd till 3rd. Output will be (786, 2.23).

### 11. What are the methods that are used in Python Tuple?

Methods that add items or remove items are not available with tuple. Only the following two methods are available:

- a) count(x)- returns the number of items that is equal to x
- b) index(x)- returns index of first item that is equal to x

### 12. Is tuple comparison possible? Explain how with example.

The standard comparisons (`_<`, `>`, `<=`, `>=`, `==`) work exactly the same among tuple objects. The tuple objects are compared element by element.

```
>>>a=(1,2,3,4,5)
>>>b=(9,8,7,6,5)
```

```
>>>a<
b True
```

### 13. What are the built-in functions that are used in Tuple?

- all()- returns true if all elements of the tuple are true or if tuple is empty
- any()- returns true if any element of tuple is true
- len()- returns the length in the tuple
- max()- returns the largest item in tuple
- min()- returns the smallest item in tuple
- sum()- returns the sum of all elements in tuple

### 14. What is the output of print tuple + tinytuple if tuple = ( 'abcd', 786 , 2.23, 'john', 70.2 ) and tinytuple = (123, 'john')?

It will print concatenated tuples. Output will be ('abcd', 786, 2.23, 'john', 70.200000000000003, 123, 'john').

### 15. Explain what is dictionary and how it is created in Python?

Dictionaries are similar to other compound types except that they can use any immutable type as an index. One way to create a dictionary is to start with the empty dictionary and add elements. The empty dictionary is denoted {}:

```
>>> eng2sp = {}
>>> eng2sp['one'] = 'uno'
>>> eng2sp['two'] = 'dos'
```

### 16. What is meant by key-value pairs in a dictionary?

The elements of a dictionary appear in a comma-separated list. Each entry contains an index and a value separated by a colon. In a dictionary, the indices are called keys, so the elements are called key-value pairs. >>> print eng2sp {'one': 'uno', 'two': 'dos'}

### 17. How does del operation work on dictionaries? Give an example.

The del statement removes a key-value pair from a dictionary. For example, the following dictionary contains the names of various fruits and the number of each fruit in stock:

```
>>> inventory = {'apples': 430, 'bananas': 312, 'oranges': 525, 'pears': 217}
```

```
>>> print inventory
```

```
{'oranges': 525, 'apples': 430, 'pears': 217, 'bananas': 312}
```

If someone buys all of the pears, we can remove the entry from the dictionary:

```
>>> del inventory['pears']
```

```
>>> print inventory
```

```
{'oranges': 525, 'apples': 430, 'bananas': 312}
```

### 18. What is meant by invocation? Where is it used and how?

**csenotescorner.blogspot** A method is similar to a function—it takes arguments and returns a value—but the

**pot.com** syntax is different. For example, the keys method takes a dictionary and returns a list of the keys that appear, but instead of the function syntax `keys(eng2sp)`, we use the method syntax `eng2sp.keys()`.

```
>>>
```

```
eng2sp.keys()
```

```
['one', 'three', 'two']
```

This form of dot notation specifies the name of the function, keys, and the name of the object to apply the function to, `eng2sp`. The parentheses indicate that this method has no parameters. A method call is called an invocation; in this case, we would say that we are invoking keys on the object `eng2sp`.

### 19. Explain values and items method used in dictionary with example.

The values method is similar; it returns a list of the values in the dictionary:

```
>>>
```

```
eng2sp.v
```

```
alues()
```

```
['uno', 'tres', 'dos']
```

The items method returns both, in the form of a list of tuples—one for each key-value pair:

```
>>> eng2sp.items()
```

```
[('one', 'uno'), ('three', 'tres'), ('two', 'dos')]
```

The syntax provides useful type information. The square brackets indicate that this is a list. The parentheses indicate that the elements of the list are tuples.

**20. What is the difference between modify and copy operations performed in dictionary?**

If you want to modify a dictionary and keep a copy of the original, use the copy method. For example, opposites is a dictionary that contains pairs of opposites:

```
>>> opposites = {'up': 'down', 'right': 'wrong', 'true': 'false'}
```

```
>>> alias = opposites
```

```
>>> copy = opposites.copy()
```

alias and opposites refer to the same object; copy refers to a fresh copy of the same dictionary. If we modify alias, opposites is also changed:

```
>>> alias['right'] = 'left'
```

```
>>> opposites['
right']
'left'
```

If we modify copy, opposites is unchanged:

```
>>> copy['right'] = 'privilege'
```

## **PART\_B**

1. Answer the following questions.

- Write short note on Functional Programming Approach. (4 marks)
- What is the difference between Lists and Tuples? Give an example for their usage. (4 marks)
- Explain the purpose of loop structure in a programming language. Describe the syntax and semantics of any two loop structures provided by Python. (8 marks)

2. Answer the following questions.

- Explain the features of a dictionary.
- What are the three types of import statement in Python? Explain.
- Write a short note on assert function.

3. Answer the following questions.

- What is the difference between lists, tuples and dictionaries? Give an example for their usage.
- What type of conditional structures are present in a programming language? How many of them are supported in Python? Explain each with example.

4. What are the basic list operations that can be performed in Python?

Explain each operation with its syntax and example.

5. What is Dictionary? Explain Python dictionaries in detail discussing its operations and methods.

## UNIT IV and V PART\_A

### FUNCTION AND FILES

#### 1. What is module and package in Python?

In Python, module is the way to structure program. Each Python program file is a module, which imports other modules like objects and attributes.

#### 2. Explain how can you access a module written in Python from C?

We can access a module written in Python from C by following method, `Module = PyImport_ImportModule(—<modulename>||);`

#### 3. Mention five benefits of using Python?

- Python comprises of a huge standard library for most Internet platforms like Email, HTML, etc.
- Python does not require explicit memory management as the interpreter itself allocates the memory to new variables and free them automatically
- Provide easy readability due to use of square brackets
- Easy-to-learn for beginners
- Having the built-in data types saves programming time and effort from declaring variables

#### 4. How to open a new file in Python?

Opening a file creates a file object. In this example, the variable `f` refers to the new file object.

```
>>> f = open("test.dat", "w")
>>> print f
<open file _test.dat, mode _w at fe820>
```

The `open` function takes two arguments. The first is the name of the file, and the second is the mode. Mode `"w"` means that we are opening the file for writing.

#### 5. Explain how the write method works on a file.

```
>>> f.write("Now is the time")
>>> f.write("to close the file")
```

Closing the file tells the system that we are done writing and makes the file available for reading:

```
>>> f.close()
```

#### 6. Which method is used to read the contents of a file which is already created?

The `read` method reads data from the file. With no arguments, it reads the entire contents of the file:

```
>>> text = f.read()
>>> print text
```

Now is the time  
to close the file

### **7. What is a text file? Give an example for a text file.**

A text file is a file that contains printable characters and whitespace, organized into lines separated by newline characters.

To demonstrate, we'll create a text file with three lines of text separated by newlines:

```
>>> f = open("test.dat", "w")
>>> f.write("line one\nline two\nline three\n")
>>> f.close()
```

### **8. What is the difference between break and continue statement?**

The break statement is new. Executing it breaks out of the loop; the flow of execution moves to the first statement after the loop.

The continue statement ends the current iteration of the loop, but continues looping.

The flow of execution moves to the top of the loop, checks the condition, and proceeds accordingly.

### **9. What is meant by directory? How and where is it useful?**

When we want to open a file somewhere else, you have to specify the path to the file, which is the name of the directory (or folder) where the file is located:

```
>>> f = open("/usr/share/dict/words", "r")
>>>
>
printf.readline()
Aarhus
```

This example opens a file named words that resides in a directory named dict, which resides in share, which resides in usr, which resides in the top-level directory of the system, called .

### **10. Explain pickling and how import pickle works.**

Pickling is so called because it —preserves data structures. The pickle module contains the necessary commands. To use it, import pickle and then open the file in the usual way:

```
>>> import pickle
>>> f = open("test.pck", "w")
```

### **11. What is an exception? Explain with few examples.**

Whenever a runtime error occurs, it creates an exception. Usually, the program stops and Python prints an error message.

Forexample,  
dividing by zero creates an exception:

```
>>> print 55/0
ZeroDivisionError: integer
division or modulo So does
accessing a nonexistent list item:
>>> a = []
>>> print a[5]
```

## 12. What is module and package in Python?

In Python, module is the way to structure program. Each Python program file is a module, which imports other modules like objects and attributes.

## 13. Explain how can you access a module written in Python from C?

We can access a module written in Python from C by following method, Module =  
=PyImport\_ImportModule(—<modulename>||);

## 14. Mention five benefits of using Python?

- Python comprises of a huge standard library for most Internet platforms like Email, HTML, etc.
- Python does not require explicit memory management as the interpreter itself allocates the memory to new variables and free them automatically Provide easy readability due to use of square brackets
- Easy-to-learn for beginners
- Having the built-in data types saves programming time and effort from declaring variables

## 15. How to open a new file in Python?

Opening a file creates a file object. In this example, the variable f refers to the new file object. >>> f = open("test.dat", "w")  
>>> print f<open file \_test.dat', mode \_w' at fe820> The open function takes two arguments. The first is the name of the file, and the second is the mode. Mode "w" means that we are opening the file for writing.

## 16. Explain how the write method works on a file.

```
>>> f.write("Now is the time")
>>> f.write("to close the file")
```

Closing the file tells the system that we are done writing and makes the file available for reading:

```
>>> f.close()
```

## 17. Which method is used to read the contents of a file which is already created?

The read method reads data from the file. With no arguments, it reads the entire contents of the file:

```
>>> text = f.read()
>>> print text
```

Now is the timeto close the file



### **18. What is a text file? Give an example for a text file.**

A text file is a file that contains printable characters and whitespace, organized into lines separated by newline characters.

To demonstrate, we'll create a text file with three lines of text separated by newlines:

```
>>> f = open("test.dat", "w")
>>> f.write("line one\nline two\nline three\n")
>>> f.close()
```

### **19. What is the difference between break and continue statement?**

The break statement is new. Executing it breaks out of the loop; the flow of execution moves to the first statement after the loop.

The continue statement ends the current iteration of the loop, but continues looping. The flow of execution moves to the top of the loop, checks the condition, and proceeds accordingly.

### **20. What is meant by directory? How and where is it useful?**

When we want to open a file somewhere else, you have to specify the path to the file, which is the name of the directory (or folder) where the file is located:

```
>>> f = open("/usr/share/dict/words", "r")
>>
>
printf.readline()
Aarhus
```

This example opens a file named words that resides in a directory named dict, which resides in share, which resides in usr, which resides in the top-level directory of the system, called .

### **21. Explain pickling and how import pickle works.**

Pickling is so called because it —preserves data structures. The pickle module contains the necessary commands. To use it, import pickle and then open the file in the usual way:

```
>>> import pickle
>>> f = open("test.pck", "w")
```

### **22. What is an exception? Explain with few examples.**

Whenever a runtime error occurs, it creates an exception. Usually, the program stops and Python prints an error message.

Forexample,

dividing by zero creates an exception:

```
>>> print 55/0
```

ZeroDivisionError: integer

division or modulo So does

accessing a nonexistent list item:

```
>>> a = []
```

```
>>> print a[5]
```

IndexError: list index out of range

Or accessing a key that isn't in the dictionary:

```
>>> b = {}
>>> print
b['what']
KeyError:
what
```

**23. List some few common Exception types and explain when they occur.**

- **ArithmeticError**- Base class for all errors that occur for numeric calculations.
- **OverflowError**- Raised when a calculation exceeds maximum limit for a numeric type.
- **ZeroDivisionError**- Raised when division or modulo by zero takes place.
  - place.
- **ImportError**- Raised when an import statement fails.
- **IndexError**- Raised when an index is not found in a sequence.
- **RuntimeError**- Raised when a generated error does not fall into any category.

**24. Write a simple program which illustrates Handling Exceptions. w**

```
x=int(input(—Please
enter a number:ℓ))
break
except ValueError:
 print(—Oops! That was no valid number. Try again...ℓ)
```

**25. What are packages in Python?**

A package is a collection of Python modules. Packages allow us to structure a collection of modules. Having a directory of modules allows us to have modules contained within other modules. This allows us to use qualified module names, clarifying the organization of our software.

**26. Explain what is meant by namespaces and scoping.**

Variables are names or identifiers that map to objects. A namespace is a dictionary of variable names/keys and their corresponding object values. A python statement can access variables in a local namespace and global namespace. If the local and global variables have the same name, the local variable shadows the global variable. Each function has its own local namespace.

## **PART B**

- 1) Answer the following questions.
  - a) Write a small code to illustrate try and except statements in Python. (4 marks)
  - b) What are packages? Give an example of package creation in Python. (4 marks)
  - c) Compare and contrast Extending and Embedding Python. (4 marks)
  - d) Write an algorithm to check whether a student is pass or fail, the total marks of student being the input. (4 marks)
- 2) Answer the following questions.
  - a) Write a program to enter a number in Python and print its octal and hexadecimal equivalent.
  - b) Demonstrate the use of Exception Handling in Python.
- 3) Answer the following questions.
  - a) What are modules in Python? Explain.
  - b) Explain in details about namespaces and scoping
  - c) Explain about the import statement in modules.
- 4) Answer the following questions.
  - a) Explain about the different types of Exceptions in Python.
  - b) Describe about Handling Exceptions in detail with examples
- 5) Explain in detail about Python Files, its types, functions and operations that can be performed on files with examples.

## UNIT VI

### OBJECT ORIENTED PROGRAMMING IN PYTHON

#### What is Object-Oriented Programming? (OOPs concepts in Python)

Object Oriented Programming is a way of computer programming using the idea of “[objects](#)” to represents data and methods. It is also, an approach used for creating neat and reusable code instead of a redundant one. The program is divided into self-contained objects or several mini-programs. Every Individual object represents a different part of the application having its own logic and data to communicate within themselves.

Now, to get a clearer picture of why we use oops instead of pop, I have listed down the differences below.

#### Difference between Object-Oriented and Procedural Oriented Programming

| Object-Oriented Programming (OOP)                                        | Procedural-Oriented Programming (Pop)                          |
|--------------------------------------------------------------------------|----------------------------------------------------------------|
| It is a bottom-up approach                                               | It is a top-down approach                                      |
| Program is divided into objects                                          | Program is divided into functions                              |
| Makes use of <i>Access modifiers</i><br>'public', 'private', 'protected' | Doesn't use <i>Access modifiers</i>                            |
| It is more secure                                                        | It is less secure                                              |
| Object can move freely within member functions                           | Data can move freely from function to function within programs |
| It supports inheritance                                                  | It does not support inheritance                                |

That was all about the differences, moving ahead let's get an idea of Python OOPs Concepts.

#### What are Python OOPs Concepts?

Major OOP (object-oriented programming) concepts in Python include Class, Object, Method, Inheritance, Polymorphism, Data Abstraction, and Encapsulation. That was all about the differences, moving ahead let's get an idea of classes and objects.

#### What are Classes and Objects?

A class is a collection of objects or you can say it is a blueprint of objects defining the common attributes and behavior. Now the question arises, how do you do that? Well, it logically groups the data in such a way that code reusability becomes easy. I can give you a real-life example- think of an office going 'employee' as a class and all the attributes related to it like 'emp\_name', 'emp\_age', 'emp\_salary', 'emp\_id' as the objects in Python. Let us see from the coding perspective that how do you instantiate a class and an object.

Class is defined under a “Class” Keyword.

**Example:**

```
1 class class1(): // class 1 is the name of the class
```

**Note:** Python is not case-sensitive.

## Objects:

Objects are an instance of a class. It is an entity that has state and behaviour. In a nutshell, it is an instance of a class that can access the data.

**Syntax:** obj = class1()

Here obj is the “object “ of class1.

## Creating an Object and Class in python:

### Example:

```
1 class employee():
2 def __init__(self,name,age,id,salary): //creating a function
3 self.name = name // self is an instance of a class
4 self.age = age
5 self.salary = salary
6 self.id = id
7
8 emp1 = employee("harshit",22,1000,1234) //creating objects
9 emp2 = employee("arjun",23,2000,2234)
10 print(emp1.__dict__)//Prints dictionary
```

**Explanation:** 'emp1' and 'emp2' are the objects that are instantiated against the class 'employee'. Here, the word (\_\_dict\_\_) is a “dictionary” which prints all the values of object 'emp1' against the given parameter (name, age, salary).(\_\_init\_\_) acts like a constructor that is invoked whenever an object is created.

## Object-Oriented Programming methodologies:

Object-Oriented Programming methodologies deal with the following concepts.

- Inheritance
- Polymorphism
- Encapsulation
- Abstraction

## UNIT – 1

1. a) What is an algorithm? Explain characteristics of an algorithm.  
b) What are the features and applications of Python?
2. Explain about the input and output statements or methods with example.
3. a) Explain about the Single-Valued data types in python.  
b) What is Indentation? Explain with example.
4. a) What is data type? List out the types of data types with example.  
b) Explain variable assignment with suitable example.  
Describe an algorithm for the following  
Prime number or not  
Odd or even
6. Define Flowchart and explain symbols used in flowchart with example.
7. Explain briefly about the Multi-Valued Data types with example.
8. a) Enumerate the list and its methods with example.  
b) Elucidate the string and its methods with example.
9. a) What is dictionary? Explain the methods available in dictionary.  
b) Differentiate between the tuple and sets in python.
10. a) Define Variable and mention rules for choosing names of Variable.  
b) Implement the python program to calculate total and average marks based on input

## **UNIT – 2**

1. List various types of operators in Python and write any 4 types of operators.
2. Explain in detail about Control flow structures in python.
3. Outline with an example the assignment and bitwise operators supported in Python.
4. a) Write a Python program to print prime number series up to N.  
b) Write a Python program to Swapping of two numbers with and without using temporary variable.  
a) If the age of Ram, Sam, and Khan are input through the keyboard, write a python program to determine the eldest and youngest of the three.  
Arithmetic operations (Addition, Subtraction, Multiplication, and Division) on integers. Input the two integer values and operator for performing arithmetic operation through keyboard
5. a) Write a python program to find the given number is odd or even.  
b) Explain break and continue statement with the help of for loop with an example.
6. Explain the syntax of the following statements
  - i) for loop
  - ii) while loop
  - iii) if - else
  - iv) if-elif-else
7. Explain expressions in python with order of evaluation with example.
8. a) Develop a program to find the largest among three numbers.  
b) Write a python program to find the given year is leap or not.
- 10.a) Write a python program to print factorial of a given number  
b) Implement Python program to find sum of natural numbers.

## UNIT – 4 and 5

1. a) Define function? Write its syntax  
b) What are formal and actual arguments  
c) Write functions to do all arithmetic operations
  2. a) What is recursive function?  
b) Explain lambda function  
c) Write recursive function to find factorial of a number
  3. a) Differentiate key word and default arguments  
b) Differentiate global and local variables
  4. a) Compare class and object with python code  
b) Narrate scope of a variable in a function
  5. a) Write about class constructor(`_init_()`),self-variable
- b) Compare method overloading and overriding
- a) A 4 digit integer is entered through the keyboard. Write a function to Calculate the sum of the 4 digit number both without recursion and using recursion
- c) What are the cons of using functions
  - d) What is inheritance? Illustrate types of inheritance with python code
- a) Write a function to display ASCII Code of entered character
- b) Describe how an object is passed as parameter to a method
- a) Write a class with following criteria
- Class name: Flower
- Objects: lilly, rose, hibiscus
- Properties: price, color,smellMethods: `get()`, `display()`
- b) Narrate about polymorphism
6. a)Write a function to return right most digit in the entered number
- b) How do we achieve code reusability in python? Write a python code to Explain code reusability



## UNIT – 4

- 1
  - a Explain in details about namespaces and scoping.
  - b Explain about the import statement in modules
- 2
  - a What are packages? Give an example of package creation in Python.
  - b Write a small code to illustrate package in Python.
- 3 What is an Exception? Explain types of exceptions and give an example.
- 4 How to create user defined exception in python with an example.
- 5
  - a What is module? How many ways to import module in python?  
Write a python code to check the given number is prime or not using modules
  - b modules
- 6
  - a What is searching? Explain with example program.
  - b What is matching? Explain with example program.
- 7
  - a List some few common Exception types and explain when they occur
  - b Difference between Raising and User defined exceptions in python?
- 8
  - a Describe any one regular expression with an example?
  - b Explain package installation via PIP
- 9 Write a syntax for Handling Exceptions with example
  - i try
  - ii try-except
  - iii try-except-else
  - iv try-except-else-finally
- 10
  - a What are modules in Python?
  - b How to create a module?
  - c What is import statement in modules?
  - d What is from import statement in modules?

## UNIT – 5

- 1 Explain in detail about Python Files, its types, functions and operations that can be performed on files with examples.
- 2
  - a Explain about command line arguments
  - b Explain about reading and writing files in python
- 3
  - a Explain about Data Compression
  - b Explain about Dates and Times
- 4
  - a What is mean by Iterators and Generators?
  - b Write about map and filter in Python?
- 5 Demonstrate about the GUI programming in Python
  - i Circle
  - ii Triangle
  - iii Rectangle
- 6
  - a What is mean by Text File? Explain about reading and writing files in python?
  - b What is turtle with an example?
- 7 Explain about Functional Programming.
  - i Iterators
  - ii Generators
  - iii Maps
  - iv Filters
- 8 Explain about Mathematics function in python
- 9
  - a Explain about Python Runtime Services
  - b Explain about Data Compression
- 10
  - a What is Data Management and Object Persistence
  - b Explain about Dates and Times

