

SEC-183: PYTHON PROGRAMMING

(Contact Hrs-60, Credits-3)

Course Objectives (COs):

To Introduce Python Programming Language as Multipurpose Programming Language with Features and Applications. This course is designed to equip students with the basic skills and knowledge of python programming.

Learning outcomes (LOs):

Students shall be able to Develop programs in Python platforms using lists, tuples and strings along with several libraries. Learn core Python structures and flow control, Create and run python function. Install and use Python on Various Platform.

UNIT-I:Introduction to basics of Python (Theory) 15 **Hours**

Introduction to Python:Python overview, Python interpreter and shell, Python identifiers, keywords, variables, standard data types, numbers(integers, floating point numbers, complex numbers), operators, statement and expression, string operations, Boolean expression, control expressions, Iteration- while statement, break, continue and pass.

UNIT-II:Function, Built in data-structures (Theory) 15 **Hours**

Functions, Built-in functions, composition of functions, user defined functions, parameters and arguments, Function calls, Recursion, stack diagrams for recursive functions, the anonymous functions, writing Python scripts,errors and exceptions. Built-in Data structures in Python: Strings, Lists, Ranges, Tuples, Dictionaries, Text files, Directories, Manipulations Building blocks of python program.

UNIT-III: Practicals 30 Hours

Suggested Practical Assignments (Questions need not be restricted to this list)

Basic Practicals:

1. Write Python program to demonstrate the use of operators and expressions
2. Write Python program to demonstrate the use of control statements
3. Write Python program to demonstrate the use of integers, floating point and complex
4. Numbers.

5. Write Python program to demonstrate the use of built-in functions
6. Write Python program to demonstrate the use of user defined functions
7. Write Python program to demonstrate the use of Lists and ranges
8. Write Python program to demonstrate the use of dictionaries
9. Write Python program to demonstrate the use of strings.
10. Find all numbers between 2000 and 2500 which are multiples of 17, but not the multiple of 5.
11. Print the first 2 and first 3 Characters in a given String using string slicing.
12. Write a program that eliminates duplicates in a list.

Advance Practicals:

1. Implement shallow copy and deep copy of a list.
2. Find the largest of n numbers, using a user defined function largest().
3. Write a function that capitalizes all vowels in a string.
4. Read a line containing digits and letters. Write a program to give the count of digits and letters.
5. Write a function *myReversal()* which receives a string as an input and returns the reverse of the string.
6. Use the list comprehension methodology in Python to generate the squares of all odd numbers in a given list.
7. Generate a dictionary and print the same. The keys of the dictionary should be integers between *L* and *LO* (both inclusive). The values should be the cubes of the corresponding keys.
8. Create a nested dictionary. The roll number of a student maps to a dictionary. This inner dictionary will have name, age and place as keys. Read details of at least three students.
9. Enter a word. Create a dictionary with the letters of this word as keys, and the corresponding ASCII values as values.
10. Write a Python function that takes a list and returns a new list with distinct elements from the list.
11. Python program to search a Word location in String.
12. Write a Python program to find repeated items in a tuple.
13. Program to concatenate the two dictionaries and create a new one.

Instructions to Paper Setter

- Questions should be set according to the following scheme.

UNIT	Questions	
	To be set	To be Answered

I	2	1
II	2	1

For Practical a total of 10 questions, each carrying 19 marks, shall be set. For each question, there shall be two sub-questions, one carrying 9 marks and the other carrying 10 marks. A student shall be allotted any one of the questions on a LOTTERY basis.

Exam Duration:

Theory	Practical
2 Hours	2 Hours

Evaluation of marks for practical Exam:

- 10% :Syntax and input/output screens
- 30% :Logic and efficiency(source code, pseudo code, and algorithm)
- 20% :Error trapping(illegalorinvalidinput,stackoverflow,underflow,insufficientphysicalmemoryetc.)
- 20% :Completion
- 20% : Result

Suggested Readings:

Text Books:

1. R.Thareja, Python Programming: Using Problem Solving Approach, First Edition, Oxford University Press, 2019.
2. E. Balaguruswamy , Introduction to Computing And Problem Solving Using Python, 1st Edition, McGraw Hill, 2016.

Reference Books:

1. S. Gowrishankar, A. Veena , Introduction to Python Programming, 1st Edition, CRC Press/Taylor & Francis, 2018.
2. J. W. Chun, Core Python Programming, Second Edition, Pearson, 2010.
3. M. C. Brown, Python: The Complete Reference, Osborne/McGraw-Hill, 4th Edition, 2018.