

INTRODUCTION TO PYTHON PROGRAMMING LAB

COURSE CODE: 20CA3106

L T P C

0 0 3 1.5

COURSE OUTCOMES:

- CO1:** Examine Python syntax and semantics and be fluent in the use of Python flow control and functions.
- CO2:** Demonstrate proficiency in handling Strings and File Systems.
- CO3:** Create, run and manipulate Python Programs using core data structures like Lists Dictionaries and use Regular Expressions.
- CO4:** Interpret the concepts of Object-Oriented Programming as used in Python.
- CO5:** Implement exemplary applications related to Network Programming, Web Services and Databases in Python.

I. Exercise programs on basic control structures & loops.

- a) Write a program for checking the given number is even or odd.
- b) Using a for loop, write a program that prints the decimal equivalents of 1/2, 1/3, 1/4 ,..... 1/10
- c) Write a program for displaying reversal of a number.
- d) Write a program for finding biggest number among 3 numbers.
- e) Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.

II. Exercise programs on operators & I/O operations.

- a) Write a program that takes 2 numbers as command line arguments and prints its sum.
- b) Implement python script to show the usage of various operators available in python language.
- c) Implement python script to read person's age from keyboard and display whether he is eligible for voting or not.
- d) Implement python script to check the given year is leap year or not.

III. Exercise programs on Python Script.

- a) Implement Python Script to generate first N natural numbers.
- b) Implement Python Script to check given number is palindrome or not.
- c) Implement Python script to print factorial of a number.
- d) Implement Python Script to print sum of N natural numbers.
- e) Implement Python Script to check given number is Armstrong or not.
- f) Implement Python Script to generate prime numbers series up to n

IV. Exercise programs on Lists.

- a) Finding the sum and average of given numbers using lists.
- b) To display elements of list in reverse order.
- c) Finding the minimum and maximum elements in the lists.

V. Exercise programs on Strings.

- a) Implement Python Script to perform various operations on string using string libraries.
- b) Implement Python Script to check given string is palindrome or not.
- c) Implement python script to accept line of text and find the number of characters, number of vowels and number of blank spaces in it.

VI. Exercise programs on functions.

- a) Define a function `max_of_three()` that takes three numbers as arguments and returns the largest of them.
- b) Write a program which makes use of function to display all such numbers which are divisible by 7 but are not a multiple of 5, between 1000 and 2000.

VII. Exercise programs on recursion & parameter passing techniques.

- a) Define a function which generates Fibonacci series up to n numbers.
- b) Define a function that checks whether the given number is Armstrong
- c) Implement a python script for Call-by-value and Call-by-reference
- d) Implement a python script for factorial of number by using recursion.

VIII. Exercise programs on Tuples.

- a) Write a program which accepts a sequence of comma-separated numbers from console and generate a list and a tuple which contains every number. Suppose the following input is supplied to the program: 34, 67, 55, 33, 12, 98. Then, the output should be: ['34', '67', '55', '33', '12', '98'] ('34','67', '55', '33', '12', '98').
- b) With a given tuple (1, 2, 3, 4, 5, 6, 7, 8, 9, 10), write a program to print the first half values in one line and the last half values in one line.

IX. Exercise programs on files.

- a) Write Python script to display file contents.
- b) Write Python script to copy file contents from one file to another.

X. Exercise programs on searching & sorting Techniques.

- a) Implement a python script to check the element is in the list or not by using Linear search & Binary search.
- b) Implement a python script to arrange the elements in sorted order using Bubble, Selection, Insertion and Merge sorting techniques.

REFERENCES:

1. <https://www.tutorialspoint.com/python/index.html>
2. <https://www.programiz.com/python-programming>
3. <https://docs.python.org/3/tutorial/>