18MC201 PROGRAMMING THROUGH PYTHON

Course Description and Objectives:

This course is aimed at offering the fundamental concepts of Python scripting language to the students. It starts with the basics of Python programming and deals with lists, dictionaries, functions, exceptions and files. The objective of this course is to enable the students to develop the applications using the concepts of Python.

Course Outcomes:

A student will be able to:
- Understand the basic terminology used in computer programming to write, compile and debug programs in Python programming language.
- Use different data types to design programs involving decisions, loops, and functions.
- Handle the exceptions which are raised during the execution of Python scripts.
- Implement files and classes in the Python programming environment.

Skills:

- Identify suitable data types for an application.
- Use control statements for decision making problems.
- Design an application to perform various operations using classes.
- Create a list of data and perform operations on data.

Activities:

- Utilization of data types such as lists, tuples, dictionaries and sets.
- Develop functions (built-in and user-defined)
- Implement file operations.
- Perform operations such as display, calculate, add, delete and modify on data from applications.
- Implement matrix operations.

Syllabus

UNIT – 1   12 Hours

INSTALLATION, DATA TYPES AND INPUT/OUTPUT**:** Importance of Python, Installing Python in Windows & Ubuntu, Executing Python programs, Comments in Python, Internal working of Python, Python character set, Tokens, Python Core Data Types, The print() function, Assignment of values to variables, The input() function, The eval() function.

UNIT – 2   12 Hours

OPERATORS AND CONTROL STATEMENTS: Operators- Arithmetic Operators, Operator precedence and Associativity, Bitwise operator, The compound assignment operator; Decision statements- Boolean operators, Boolean Expressions and Relational operators, Decision making statements; Loop Control Statements-while loop, range() function, for loop; break statement, continue statement.

UNIT – 3   12 Hours

FUNCTIONS AND LISTS:Functions- Syntax and basics of a function, Use of a function, Parameters and arguments in a function, The local and global scope of a variable, The return statement, Recursive functions, The lambda function; Lists-Creating Lists, Accessing the elements of a List, List slicing, Python in-built functions for lists, List Comprehension, List Methods, Passing list to a function, Returning a list to function.

UNIT – 4   12 Hours

TUPLES, SETS AND DICTIONARIES:Tuples **-** Creating tuples, tuple() function, Inbuilt functions for tuples, Indexing and Slicing, Operations on tuples, Passing variable length arguments to tuples, Sort tuples, Traverse tuples from a list, The zip() function, The Inverse zip(*) function; Sets - Creating sets, The set in and not in operator, The Python Set Class, Set operations; Dictionaries -Basics of Dictionaries, Creating a Dictionary, Adding and replacing values, Retrieving values, Formatting dictionaries, Deleting items, Comparing two dictionaries, Methods of dictionary class, Traversing dictionaries, Nested dictionaries, Traversing nested dictionaries.

UNIT – 5   12 Hours

FILES AND OBJECT-ORIENTEDENVIRONMENT:File Handling**-**Opening a file, Writing Text, Closing files, Writing numbers to a file, Reading Text, Reading numbers from a file, Appending data, seek() function; Object Oriented Programming **-** Defining Classes, The self parameter and adding methods to a class, Display class attributes and methods, The __init__ Method(Constructor), __del__() (Destructor Method), Inheritance, Types of Inheritance, Using super(), Method Overriding.

LIST OF EXPERIMENTS:

Exercise 1 – Basics
a) Running instructions in Interactive interpreter and a Python Script.
b) Write a program to purposefully raise Indentation error and correct it.
Exercise 2 – Operations
a) Write a program to compute distance between two points taking input from the user   (Pythagorean Theorem).
b) Write a program add.py that takes 2 numbers as command line arguments and prints its sum.
c) Read two integers and perform all arithmetic operations on those two numbers.
Exercise - 3 Control Flow
a) Write a program for checking whether the given number is an even number or not.

b) Using a for loop, write a program that prints out the decimal equivalents of 1/2, 1/3, 1/4, . . . ,1/10.

c) Write a program using for loop that loops over a sequence.

d) Write a program using a while loop that asks the user for a number, and prints acountdown from that number to zero.

Exercise 4 - Control Flow – Continued

a) Find the sum of all the primes up to given range.

b) Each new term in the Fibonacci sequence is generated by adding the previous two terms. Starting with 1 and 2, the first 10 terms will be: 1, 2, 3, 5, 8, 13, 21, 34, 55, 89.

c) By considering the terms in the Fibonacci sequence whose values do not exceed one lakh, find the sum of the even-valued terms.

Exercise - 5 Dictionary Data Structure

a) Write a program to count the numbers of characters in the string and store them in a dictionary data structure.

b) Write a program to use split and join methods in the string and trace a birthday with a dictionary data structure.

c) Create a dictionary of student data and search for a student.

Exercise - 6 Dictionary Data Structure

a) Write a program combine lists that combines these lists into a dictionary.

b) Write a program to count frequency of characters in a given file. Can you use character frequency to tell whether the given file is a Python program file, C program file or a textfile.

Exercise - 7 Files

a) Write a program to print each line of a file in reverse order.

b) Write a program to compute the number of characters, words and lines in a file.

c) Handle divides by zero exception.

Exercise - 8 Functions

a) Write a function ball collides that takes two balls as parameters and computes if they are colliding. Your function should return a Boolean representing whether or not the balls are colliding.

Hint: Represent a ball on a plane as a tuple of (x, y, r), r being the radius

If (distance between two balls centers) <= (sum of their radii) then (they are colliding).

b) Find mean, median, mode for the given set of numbers in a list.

Exercise - 9 Functions

a) Write a function nearly equal to test whether two strings are nearly equal. Two strings a and b are nearly equal when a can be generated by a single mutation on b.

b) Write a function **dups** to find all duplicates in the list.

c) Write a function **unique** to find all the unique elements of a list.

Exercise - 10 Functions - Problem Solving

a) Write a function cumulative product to compute cumulative product of a list of numbers.

b) Write a function reverse to reverse a list. Without using the reverse function.

c) Write function to compute GCD, LCM of two numbers. Each function shouldn't exceed one line.

Exercise 11 Multi-Dimensional Lists

a) Write a program that defines a matrix and prints

b) Write a program to perform addition of two square matrices

c) Write a program to perform multiplication of two square matrices

Exercise - 12 OOP

a) Class variables and instance variable

   i) Robot

   ii) ATM Machine

   iii) Create a class called Student and perform operations such as display,
      Calculate percentage, add, delete and modify student data.

Text Book:

Ashok Namdevkamthane and Amit Ashok Kamthane, "Programming and Problem solving with PYTHON", 1st Edition, McGraw Hill Education, 2016.

Reference Books:

1. Allen Downey, "Think Python", 1st Edition, Green Tea Press, 2016.
2. W.J. Chun, "Core Python Programming", 3rd Edition, Prentice Hall, 2013.
3. Kenneth A. Lambert, "Fundamentals of Python", 2st Edition, Cengage, 2015.