ITM (SLS) Baroda University
School of Computer Science, Engineering and Technology
Syllabus
Semester 2

Course Name: Programming in Python-II
Course Code: C22X0C4
Course Type: Core

Lab Experiments:

1. Create a Python class named Calculator constructed by num1 and num2 and four methods to find addition, subtraction, ,multiplication and division.

   Take different instances (objects) of class Rectangle and apply the above methods.

2. Create a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle.

   Take different instances (objects) of class Rectangle and apply the above methods.

3. Create a Python class named Rectangle constructed by a length and width and a method which will compute the area of a rectangle.
   Take different instances (objects) of class Rectangle and apply the above methods.

4. Create a Python class called BankAccount which represents a bank account, having as attributes: accountNumber (numeric type), name (name of the account owner as string type), balance.Create a constructor with parameters: accountNumber, name, balance. Create a Deposit() method which manages the deposit actions. Create a Withdrawal() method which manages withdrawals actions. Create an bankFees() method to apply the bank fees with a percentage of 5% of the balance account. Create a display() method to display account details.

5. Create a class, Triangle. Its __init__() method should take self, angle1, angle2, and angle3 as arguments. Create a variable named number_of_sides and set it equal to 3. Create a method named check_angles and in that print the sum of three angles of traingle. It should return True if the sum of self.angle1, self.angle2, and self.angle3 is equal 180, and False otherwise. Create a variable named my_triangle and set it equal to a new instance of your Triangle class. Pass it three angles that sum to 180 (e.g. 90, 30, 60). Print out my_triangle.number_of_sides and my_triangle.check_angles().

6. Create a class, Triangle. Its __init__() method should take self, side1, side2, and side3 as arguments. Create another class Type and create a method type_of_triangle which will check for:

   a) An equilateral triangle is a triangle in which all three sides are equal.

b) A scalene triangle is a triangle that has three unequal sides.

c) An isosceles triangle is a triangle with (at least) two equal sides.

7. Write a Python program to perform single level and hierarchal inheritance in python. Declare class Student define attributes: enrollmentNo, discipline, gotMarks. Derive class Graduate and class Postgraduate from class Student. Define methods: percentage() and result() in class Student, class Graduate, and class Postgraduate. Class Student will use default values for maxMarks = 500, and passPercentage = 35%; for class Graduate, maxMarks = 600, and passPercentage = 40%; for class Postgraduate, maxMarks = 400, and passPercentage = 50%. Also, measure the time and space complexity of the program.
8. Define class Shape by inheriting class Circle, class Rectangle, and class Triangle(area=1/2 x b x h, perimeter = base+height+hypotenuse). Redefine methods: perimeter() and area() for class Shape using the concept of abstraction.
9. Write a Python program to demonstrate multilevel and hierarchical inheritance.

10. Write a Python program that accepts two numbers from the user and performs their division and if the divisor is zero then raise ZeroDivisionError. The code should demonstrate the uses of else and finally blocks as well.

11. Create a user defined exception for a salary entered by user should be in range 5000 to 10000.

12. Create a Matrix class and overload the addition, subtraction and multiplication operator for performing addition, subtraction and multiplication on two matrices.

13. Create a custom exception for the above class in a way to create Binary Matrix only. (Binary matrix is a matrix is one in which all the elements are either 0 or 1). Throw an exception when user enters any value other than 0 or 1 for any element of Matrix.

14. Write a Python program to raise an exception of IndexError python.

15. Write a Python program to demonstrate the concept of multithreading in python.

16. Write a Python program to create two functions which will calculate the cube and square of any given number and should apply the concept of multithreading with join, start and thread information of both the functions.