

# **Object Oriented Programming using Java**

## **Assignment - 4**

### **'this', 'static', Constructor, Method Overloading**

1. Design an Employee class with instance variables id, name, and salary, and write a constructor that accepts parameters with these exact same names. Inside the constructor, use the **this** keyword to correctly assign the input parameters to the instance variables to resolve the variable shadowing conflict. Create a method to display the employee's details.
2. Write a java program which has a class BankAccount having data members accountHolderName, acNum and acBalance. The class should also contain methods deposit(), withdraw() and display the account details. Use constructor to initialize the data members of the class.
3. Create a class representing a student with fields for name, age, and roll number. Create a non-parameterized constructor to set some default values to the name, age and roll. Define a parameterized constructor to set these values when a new student object is created. Use a static variable to keep count of number of student objects are created and display it using a method. Initialize multiple students to demonstrate the use of different constructor.
4. Develop a Student class that automatically assigns a unique, sequential roll number to every new student using a combination of static and instance variables. Declare a static variable nextId initialized to 100 and instance variables for name and rollNumber. In the constructor, assign the rollNumber using the current value of nextId, then immediately increment nextId so that the next object created gets a new number. Verify the logic by creating two students, "Alice" and "Bob", ensuring Alice gets ID 100 and Bob automatically gets ID 101.
5. Write a class Student that has two overloaded constructors. One constructor should accept only the student name. Another constructor should accept the student name and age. Form the second constructor call the first constructor to initialize the name. Demonstrate the usage of both constructors in the main method.
6. Write a class AreaCalculator that contains overloaded methods named calculateArea. One method should calculate the area of a square (using one side length), another method should calculate the area of a rectangle (using length and width), and another method should calculate the area of a circle (using the radius).
7. Create a class MaxFinder that contains three overloaded methods named findMax. Each method should take a different type of input. One method should find the maximum of two integers. One method should find the maximum of two floats. One method should find the max element out of an array. Demonstrate the usage of these overloaded methods in the main method.

8. Write a class to represent a point in 2D space. Implement a constructor that takes x and y coordinates. Use the 'this' keyword to differentiate between instance variables and constructor parameters. Define a static method in the class to find the distance between two points.
9. Design a class to represent a complex number. Define two methods to perform addition and subtraction between two complex number. Use constructor to initialize the complex number. Use suitable method to display a complex number.
10. Implement a java class to represent time in Hour, minute and seconds format. Consider the time is represented in 24 hour time format. Use parameterized constructor to initialize the time. Use suitable methods to add to add two times and display time in suitable format.