**Problem Set – 4**

1. Create a Student class with attributes for name and age. Implement a default constructor to assign default values and a parameterized constructor to initialize the attributes with user-defined values. Create objects using both constructors and display their details.
2. Create a BankAccount class with a private variable balance to store the account balance. Implement a public method deposit(double amount) to add funds, a protected method withdraw(double amount) to deduct funds, and a default-access method checkBalance() to display the current balance. Create an object of the class and demonstrate which methods and variables can be accessed both inside and outside the class.
3. Create a Calculator class that contains a method add() to perform addition. Overload the add() method to handle different types and numbers of parameters, such as adding two integers, two doubles, and three integers. Create an object of the class and demonstrate all method variations.
4. Create a University class that has a static variable universityName and a non-static variable studentName. Include a static method to display the university name. Then, create multiple student objects to demonstrate how the static variable is shared among all instances, while the non-static variable holds unique values for each object.
5. A student is developing a course registration system that allows students to enroll in courses. Each course has a course name and a course code. Implement a Course class with appropriate attributes and use the “this” keyword to differentiate between class attributes and constructor parameters during initialization. Create an object of the Course class and display the course details.

Additional Question:

1. A company wants to develop an Employee Management System to track employee details such as name, department, salary, and employee ID. The system should also calculate the total salary expenditure and keep a record of the total number of employees. Implement a Java program by creating an Employee class that includes instance variables for employee ID, name, department, and salary. The class should have a default constructor that initializes employee details with default values and a parameterized constructor that sets employee details based on user input. Use a static variable totalEmployees to track the total number of employees and implement a static method to display this count. Additionally, define a method calculateSalary() that returns the salary of the employee and another method displayEmployeeInfo() to display all employee details. To ensure data encapsulation, mark the salary variable as private and provide a public method to access it. Declare the totalEmployees variable as static so that it is shared among all instances. In the main method, create multiple Employee objects using both default and parameterized constructors. Use the “this” keyword in the constructors to distinguish between class variables and constructor parameters. Finally, display the total number of employees and the salary details for each employee. The program should successfully demonstrate the behavior of static and non-static members, the initialization of objects using constructors, and the role of access modifiers in an employee management scenario.]