Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)
Department of Computer Science & Engineering

Object Oriented Programming Laboratory (CSL37)

Semester:III **Week #: 03** **Section:A,B,C**

Classes, objects, static variables

1. **Singleton Design Pattern:**
    - Explain the purpose of the Singleton design pattern. Why might you want to ensure that only one instance of a class is created?
    - Describe the implementation of the Singleton class and how it prevents multiple instances from being created.
    - Discuss potential use cases where the Singleton pattern is beneficial.
2. **Bank Account Management:**
    - Create a new bank account using the **Account** class and perform a series of deposits and withdrawals. Display the account details after each transaction.
    - Explain the role of static variables in the **Account** class. How do they contribute to tracking total accounts and total balance across all accounts?
    - Implement a method in the **Account** class to display the total balance across all accounts.
3. **Employee Management System:**
    - Instantiate several **Employee** objects and assign them to different departments within a Company. Display the total number of employees and total salary expenses.
    - Discuss the significance of static variables in the **Employee** class. How do they help in maintaining a count of total employees and tracking salary expenses?
    - Implement methods in the **Employee** class to display the total number of employees and total salary expenses.

1.

```
public class Logger {
    private static Logger instance;

    // Private constructor to prevent instantiation from outside the class
    private Logger() {
    }

    // Provide a global point of access to the instance
    public static Logger getInstance() {
        if (instance == null) {
            instance = new Logger();
        }
        return instance;
    }

    // Log a message
    public void log(String message) {
        System.out.println("Log: " + message);
        // Additional logging logic can be added here
    }
}

public class Main {
    public static void main(String[] args) {
        // Get the logger instance
        Logger logger = Logger.getInstance();

        // Use the logger to log messages
        logger.log("This is a log message.");
        logger.log("Another log message.");

        // Even if you try to create a new instance, you'll get the same one
        Logger anotherLogger = Logger.getInstance();
        System.out.println(logger == anotherLogger); // This will print true
    }
}
```

2.
```
public class Account {
    private static int totalAccounts = 0;
    private static double totalBalance = 0;

    private int accountNumber;
    private double balance;

    public Account(double initialBalance) {
        accountNumber = ++totalAccounts;
        balance = initialBalance;
        totalBalance += initialBalance;
    }

    public void deposit(double amount) {
        balance += amount;
        totalBalance += amount;
    }

    public void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            totalBalance -= amount;
        } else {
            System.out.println("Insufficient funds!");
        }
    }

    public static void displayTotalBalance() {
        System.out.println("Total Balance across all accounts: $" + totalBalance);
    }

    // Other methods can be added here
}
```

3.

```
public class Employee {
    private static int totalEmployees = 0;
    private static double totalSalaryExpenses = 0;

    private int employeeId;
    private String name;
    private double salary;

    public Employee(String name, double salary) {
        this.name = name;
        this.salary = salary;
        employeeId = ++totalEmployees;
        totalSalaryExpenses += salary;
    }

    public static void displayTotalEmployees() {
        System.out.println("Total Number of Employees: " + totalEmployees);
    }

    public static void displayTotalSalaryExpenses() {
        System.out.println("Total Salary Expenses: $" + totalSalaryExpenses);
    }

    // Other methods can be added here
}
```