

Note:

- This assignment is designed to practice static fields, static initializers, and static methods.
- Understand the problem statement and use static and non-static wisely to solve the problem.
- Use constructors, proper getter/setter methods, and toString() wherever required.
- Design and implement a class named InstanceCounter to track and count the number of instances created from this class.

-->>

```
package org.example.q1;
```

```
public class InstanceCounter {
```

```
    public static int count = 0;
```

```
    public InstanceCounter() {  
        count++;  
    }
```

```
    public static int getCount() {  
        return count;  
    }
```

```
    public void printCount() {  
        System.out.println("The number of instances are :" +  
getCount());  
    }
```

```

    }

    public static void main(String[] args) {
        InstanceCounter ic1 = new InstanceCounter();
        InstanceCounter ic2 = new InstanceCounter();
        InstanceCounter ic3 = new InstanceCounter();

        ic1.printCount();
    }
} }

}

```

- Design and implement a class named **Logger** to manage logging messages for an application. The class should be implemented as a singleton to ensure that only one instance of the **Logger** exists throughout the application.

The class should include the following methods:

- **getInstance()**: Returns the unique instance of the **Logger** class.
- **log(String message)**: Adds a log message to the logger.
- **getLog()**: Returns the current log messages as a **String**.
- **clearLog()**: Clears all log messages.

```

-->>
package org.example.q1;

```

```

class Logger {
    private static Logger instance;

    private StringBuilder logMessages = new
StringBuilder();

    private Logger() {
    }

    public static Logger getInstance() {
        if (instance == null) {
            instance = new Logger();
        }
        return instance;
    }

    public void log(String message) {
        logMessages.append(message).append("\n");
    }

    public String getLog() {
        return logMessages.toString();
    }

    public void clearLog() {

```

```

        logMessages.setLength(0);
    }
}

public class Main {
    public static void main(String[] args) {
        Logger logger = Logger.getInstance();
        logger.log("Application started.");
        logger.log("An error occurred.");

        System.out.println("Current Logs:\n" +
logger.getLog());

        logger.clearLog();

        System.out.println("Logs cleared.\nCurrent
Logs:\n" + logger.getLog());
    }
}

```

- Design and implement a class named Employee to manage employee data for a company. The class should include fields to keep track of the total number of employees and the total salary expense, as well as individual employee details such as their ID, name, and salary.

The class should have methods to:

- Retrieve the total number of employees

(getTotalEmployees())

- Apply a percentage raise to the salary of all employees (applyRaise(double percentage))
- Calculate the total salary expense, including any raises (calculateTotalSalaryExpense())
- Update the salary of an individual employee (updateSalary(double newSalary))

Understand the problem statement and use static and non-static fields and methods appropriately. Implement static and non-static initializers, constructors, getter and setter methods, and a toString() method to handle the initialization and representation of employee data.

Write a menu-driven program in the main method to test the functionalities.

-->>

```
package org.example.q1;
```

```
public class Employee {  
    private int employeeId;  
    private String name;  
    private double salary;  
  
    private static int totalEmployees = 0;  
    private static double totalSalaryExpense = 0.0;  
  
    public Employee(int employeeId, String name, double  
salary) {  
        this.employeeId = employeeId;  
        this.name = name;  
        this.salary = salary;  
        totalEmployees++;  
    }  
}
```

```

        totalSalaryExpense += salary;
    }

    public int getEmployeeId() {
        return employeeId;
    }

    public String getName() {
        return name;
    }

    public double getSalary() {
        return salary;
    }

    public static int getTotalEmployees() {
        return totalEmployees;
    }

    public static double getTotalSalaryExpense() {
        return totalSalaryExpense;
    }

    public void updateSalary(double newSalary) {
        totalSalaryExpense -= salary;
        totalSalaryExpense += newSalary;
        this.salary = newSalary;
    }

    public static void applyRaise(double percentage) {
        double raiseAmount = totalSalaryExpense *
(percentage / 100);
        totalSalaryExpense += raiseAmount;
    }

    public static double calculateTotalSalaryExpense()
{
        return totalSalaryExpense;
    }

```

```

    }

    @Override
    public String toString() {
        return "Employee ID: " + employeeId + ", Name: " + name + ", Salary: $" + salary;
    }

    public static void main(String[] args) {

        Employee emp1 = new Employee(101, "Bill
Gates", 50000.0);
        Employee emp2 = new Employee(102, "Steve
Jobs", 60000.0);

        Employee.applyRaise(25);

        emp1.updateSalary(75000.0);

        System.out.println("Total Employees: " +
Employee.getTotalEmployees());
        System.out.println("Total Salary Expense: $" +
Employee.calculateTotalSalaryExpense());

        System.out.println(emp1);
        System.out.println(emp2);
    }
}

```