

1. Design and implement a class named `InstanceCounter` to track and count the number of instances created from this class.

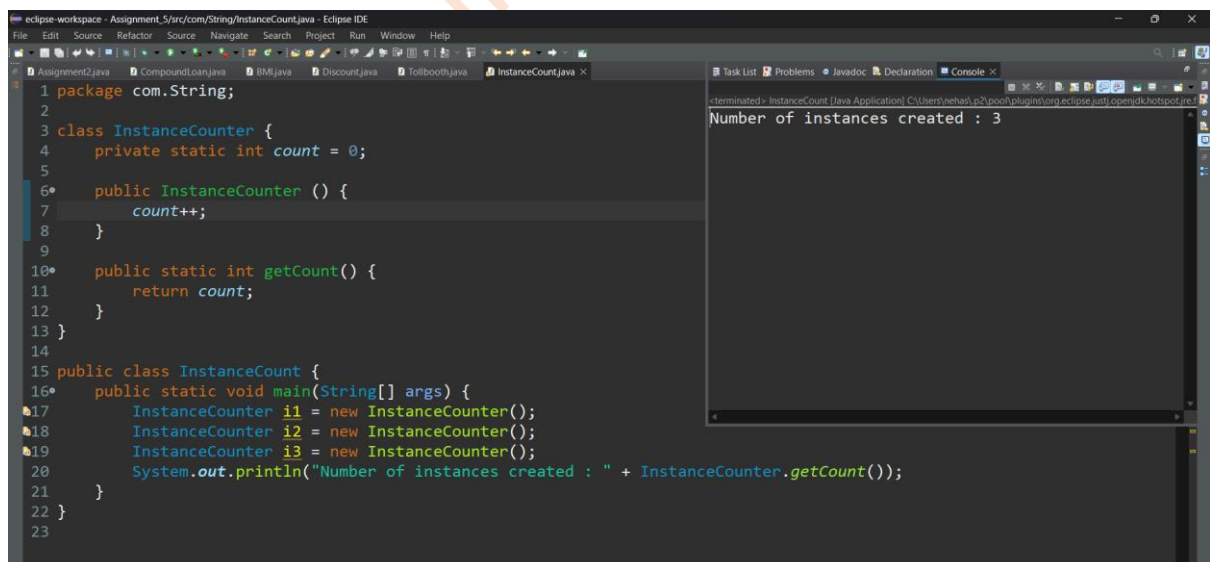
```
package com.String;
```

```
class InstanceCounter {
    private static int count = 0;

    public InstanceCounter () {
        count++;
    }

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

public class InstanceCount {
    public static void main(String[] args) {
        InstanceCounter i1 = new InstanceCounter();
        InstanceCounter i2 = new InstanceCounter();
        InstanceCounter i3 = new InstanceCounter();
        System.out.println("Number of instances created : " +
InstanceCounter.getCount());
    }
}
```



The screenshot shows the Eclipse IDE with the following code in the editor:

```
1 package com.String;
2
3 class InstanceCounter {
4     private static int count = 0;
5
6     public InstanceCounter () {
7         count++;
8     }
9
10    public static int getCount() {
11        return count;
12    }
13 }
14
15 public class InstanceCount {
16     public static void main(String[] args) {
17         InstanceCounter i1 = new InstanceCounter();
18         InstanceCounter i2 = new InstanceCounter();
19         InstanceCounter i3 = new InstanceCounter();
20         System.out.println("Number of instances created : " + InstanceCounter.getCount());
21     }
22 }
23
```

The console output on the right shows:

```
<terminated> InstanceCount [Java Application] C:\Users\neha\p2\pool\plugins\org.eclipse.jdt.core\bin\java.exe
Number of instances created : 3
```

2. 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.

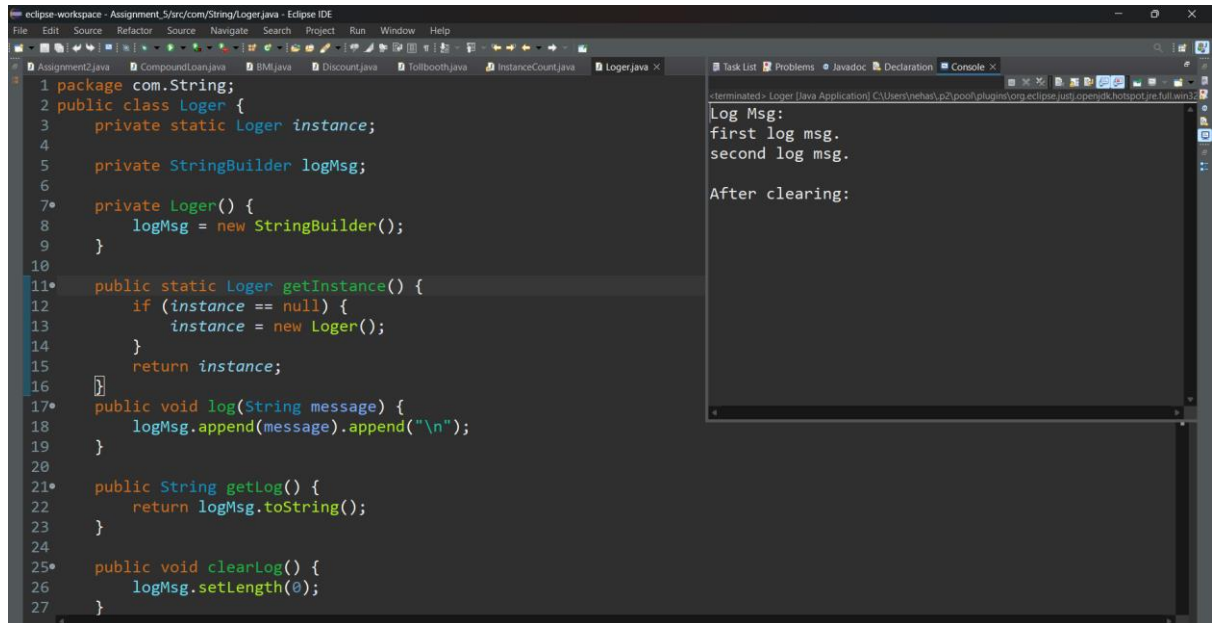
```
public class Logger {  
  
    private static Logger instance;  
  
    private StringBuilder logMsg;  
  
    private Logger() {  
        logMsg = new StringBuilder();  
    }  
  
    public static Logger getInstance() {  
        if (instance == null) {  
            instance = new Logger();  
        }  
        return instance;  
    }  
  
    public void log(String message) {  
        logMsg.append(message).append("\n");  
    }  
  
    public String getLog() {
```

```
        return logMsg.toString();
    }

    public void clearLog() {
        logMsg.setLength(0);
    }

    public static void main(String[] args) {
        Logger l = Logger.getInstance();
        l.log("first log msg.");
        l.log("second log msg.");
        System.out.println("Log Msg:");
        System.out.println(l.getLog());
        l.clearLog();
        System.out.println("After clearing:");
        System.out.println(l.getLog());
    }
}
```

ASSIGNMENT NO.6



The screenshot shows the Eclipse IDE with a project named 'Assignment_5/src/com/String/Logger.java'. The code in the editor is as follows:

```
1 package com.String;
2 public class Logger {
3     private static Logger instance;
4
5     private StringBuilder logMsg;
6
7     private Logger() {
8         logMsg = new StringBuilder();
9     }
10
11    public static Logger getInstance() {
12        if (instance == null) {
13            instance = new Logger();
14        }
15        return instance;
16    }
17    public void log(String message) {
18        logMsg.append(message).append("\n");
19    }
20
21    public String getLog() {
22        return logMsg.toString();
23    }
24
25    public void clearLog() {
26        logMsg.setLength(0);
27    }
28 }
```

The console output on the right shows the following messages:

```
Log Msg:
first log msg.
second log msg.

After clearing:
```

3. 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 com.String;
import java.util.Scanner;
class Empl{
    private static int totalEmp = 0;
    private static double totalExp = 0.0;

    private int empId;
    private String name;
    private double salary;
```

```

        {
            this.empid = ++totalEmp;
        }

        public static int getTotalEmp() {
            return totalEmp;
        }

        public static double getTotalExp() {
            return totalExp;
        }

        public void applyRaise(double percentage) {
            double r = salary * (percentage / 100);
            totalExp = totalExp + r;
            salary = salary + r;
        }

        public void updateSalary(double newSalary) {
            totalExp -= salary;
            salary = newSalary;
            totalExp = totalExp + newSalary;
        }

        public String toString() {
            return "Name : " + name + "\n" +
                "Empid : " + empid + "\n" +
                "salary : " + salary + "\n";
        }

        public void menuList() {
            System.out.println("Enter 1. Add Employee");
            System.out.println("    2. Update Salary");
            System.out.println("    3. Apply Raise to All");
            System.out.println("    4. Total Employees");
            System.out.println("    5. Total Salary Expense");
            System.out.println("    6. Exit");
        }
    }

    public class employee {
        public static void main(String[] args) {
            Empl e = new Empl();
            Scanner sc = new Scanner(System.in);
            int empCount = 0;
            boolean exit = false;

```

```

while(exit != true) {
    e.menuList();
    System.out.print("Enter your choice : ");
    int choice = sc.nextInt();

    switch(choice) {
        case 1:
            System.out.print( "Enter name : ");
            sc.nextLine();
            String name = sc.nextLine();
            System.out.println("Name : " + name);
            break;
        case 2:
            System.out.print("Enter salary: ");
            double salary = sc.nextDouble();
            System.out.println("salary : " + salary);
            break;
        case 3:
            System.out.print("Enter raise percentage: ");
            double percentage = sc.nextDouble();
            break;
        case 4:
            System.out.println("Total employees : " +
e.getTotalEmp());
            break;
        case 5:
            System.out.println("Total salary : " + e.getTotalExp());
            break;
        case 6:
            exit = true;
            break;
        default:
            System.out.println("Invalid input!");
    }
    }
    sc.close();
}
}

```

ASSIGNMENT NO.6

```
eclipse-workspace - Assignment_5/src/com/String/employee.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Assignment2.java CompoundLoan.java BMI.java Discount.java Tollbooth.java InstanceCount.java Logger.java empl... Task List Problems Javadoc Declaration Console
1
2 package com.String;
3 import java.util.Scanner;
4 class Empl{
5     private static int totalEmp = 0;
6     private static double totalExp = 0.0;
7
8     private int empId;
9     private String name;
10    private double salary;
11
12*   {
13       this.empId = ++totalEmp;
14   }
15
16*   public static int getTotalEmp() {
17       return totalEmp;
18   }
19
20*   public static double getTotalExp() {
21       return totalExp;
22   }
23
24*   public void applyRaise(double percentage) {
25       double r = salary * (percentage / 100);
26       totalExp = totalExp + r;
27       salary = salary + r;

```

terminated - employee [Java Application] C:\Users\nehas.p2\poo\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.w...
Enter 1. Add Employee
2. Update Salary
3. Apply Raise to All
4. Total Employees
5. Total Salary Expense
6. Exit
Enter your choice : 1
Enter name : Neha
Name : Neha
Enter 1. Add Employee
2. Update Salary
3. Apply Raise to All
4. Total Employees
5. Total Salary Expense
6. Exit
Enter your choice : 4
Total employees : 1
Enter 1. Add Employee
2. Update Salary
3. Apply Raise to All
4. Total Employees
5. Total Salary Expense
6. Exit
Enter your choice : 6