

5

Assignment 5 : Sandeep Sir

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

```
package com.example;

class Helper {
    private static int count;

    Helper() {
        count++;
    }

    static int getCount() {
        return count;
    }
}

public class InstanceCounter {

    public static void main(String[] args) {

        Helper util = new Helper();
        Helper util1 = new Helper();
        Helper util2 = new Helper();
        Helper util3 = new Helper();
        Helper util4 = new Helper();
        System.out.println(Helper.getCount());
    }
}
```

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.

```
package com.example;

//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 thro
//
//The class should include the following methods:
//log(String message): Adds a log message to the logger.
//getLog(): Returns the current log messages as a String.
//clearLog(): Clears all log messages.

class LoggerHelper {
    private String message;
    static LoggerHelper lh;
    LoggerHelper(){
        this.message = "";
    }

    public static LoggerHelper getInstance() {
        if (lh == null) {
            lh = new LoggerHelper();
        }
    }
}
```

```

        return lh;
    }
    return lh;
}

public void clearMessage() {
    this.message = " ";
}

public void setMessage(String msg) {
    this.message = message + " \n " + msg;
}

public String getMessage() {
    return this.message;
}
}

public class Logger {
    public static void main(String[] args) {
        LoggerHelper lh1 = LoggerHelper.getInstance();

        lh1.setMessage("My name is Aditi Mehre");
        lh1.setMessage("My name is Kiara Mehre");
        System.out.println(lh1.getMessage());
        LoggerHelper lh2 = LoggerHelper.getInstance();
        System.out.println(lh2.getMessage());
        lh1.clearMessage();

    }
}

```

3. Employee Class

```

package com.example;

import java.util.Scanner;

//
/////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 exper
//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.
class EmpHelper {
    private int totalEmp;
    private double SalaryExp;
}

```

```

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

EmpHelper(int id, String name, double salary) {
    this.empId = id;
    this.name = name;
    this.salary = salary;

    totalEmp++;
    SalaryExp = SalaryExp + salary;
}

public int getTotalEmployees() {
    return this.totalEmp;
}

public void setSalary(double sal) {
    this.salary = sal;
}

public double getSalary() {
    return this.salary;
}

public double calculateTotalSalaryExpense() {
    return this.SalaryExp;
}

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

public double updateSalary(double newSalary) {
    SalaryExp = salary - SalaryExp;
    return this.salary = SalaryExp;
}

public String toString() {
    String str = "id=" + empId + ", name=" + name + '\'' + ", salary=" + salary + ", totalSalaryExpense"
        + SalaryExp;
    return str;
}
}

public class Emp {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        EmpHelper eh = null;

        while (true) {

            System.out.println("Add Employee => Press 1");
            System.out.println("Apply Raise => Press 0");

```

```

System.out.println("Calculate Total Salary Exp => Press 3");
System.out.println("Update salary => Press 4");
System.out.println(" View Emp details  => Press 5");
System.out.println("Print salary => Press 6");
System.out.println("Exit => Press 0");

int choice = sc.nextInt();
switch (choice) {
case 1:
    System.out.println("Enter the emp id : ");
    int id = sc.nextInt();

    System.out.println("Enter the emp name : ");
    String name = sc.nextLine();

    sc.nextLine(); // Consume the newline character left by nextInt() to avoid issues in the ne
                  // input
    System.out.println("Enter the emp current salary : ");
    double salary = sc.nextDouble();

    eh = new EmpHelper(id, name, salary);
    System.out.println("Added successfully");
    break;

case 2:
    if (eh != null) {
        System.out.println(" Enter the salary raise : ");
        double percenatge = sc.nextDouble();
        eh.applyRaise(percenatge);
        System.out.println("Successfully raised applied ");

    } else {
        System.out.println("Please add employee first : ");
    }

    break;

case 3:
    if (eh != null) {
        System.out.println("Total salary expense: " + eh.calculateTotalSalaryExpense());

    } else {
        System.out.println("Please add employee first : ");
    }
    break;

case 4:
    if (eh != null) {
        System.out.println("Enter new Salary : ");
        double sal = sc.nextDouble();
        eh.updateSalary(sal);
        System.out.println("Successfully updated. ");
    } else {
        System.out.println("Please add employee first : ");
    }

    break;

case 5:
    if (eh != null) {

```

```
        System.out.println(eh.toString());
    } else {
        System.out.println("No employee added yet : ");
    }
    break;

case 6:
    System.out.println("Exiting...");
    System.exit(0);
default:
    System.out.println("Invalid choice.");
}
}
}
```