Name:117_Swapnil Dhotre_KH

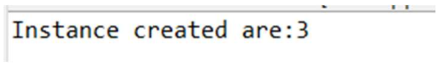
OOPJ Assignment 5

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

```
  package ass5.exapmle;
public class InstanceCounter {
	static int count;
	public InstanceCounter() {
	 count++;
	}
	public static int getInstance()
	{
		return count;
	}

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		InstanceCounter i1=new InstanceCounter();
		InstanceCounter i2=new InstanceCounter();
		InstanceCounter i3=new InstanceCounter();
		System.out.println("Instance created 
are:"+InstanceCounter.getInstance());
		}

}
```



```
Instance created are: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

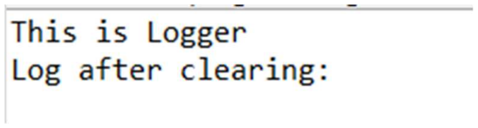
```
   package org.programming5_2;
import java.util.Scanner;



class Logger{
private String log;

private Logger(){
this.log="";
}
private static Logger reference;
public static Logger getInstance() {
 Logger reference=new Logger();
return reference;
}
public void setLog(String log) {
this.log = log;
}
public String getLog()
{
return this.log;
}
public void clearLog() {
log="";
}
}
public class programming{

public static void main(String[] args) {
        // TODO Auto-generated method stub
Logger l=Logger.getInstance();
l.setLog("This is Logger");
System.out.println(l.getLog());
l.clearLog();
System.out.println("Log after clearing:");
}
}
```



```
This is Logger
Log 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.