***Assignment – 5***

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

Ans = public class InstanceCounter {

private static int *count*;

public void printRecord() {

*count*+=1;

System.***out***.println("Count is "+*count*);

}

public static void main(String[] args) {

// **TODO** Auto-generated method stub

InstanceCounter ic = new InstanceCounter();

ic.printRecord();

ic.printRecord();

ic.printRecord();

}

}

1. 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.

**Ans =**

package instanceee;

public class Logger {

private static Logger *instance*;

private Logger() {

}

public static Logger getInstance() {

if (*instance* ==null) {

*instance* = new Logger();

}

return *instance*;

}

public void log(String message) {

System.***out***.println(message);

}

String getLog(String message) {

return message;

}

void clearLog() {

}

public static void main(String[] args) {

// **TODO** Auto-generated method stub

Logger l = new Logger();

l.*getInstance*();

l.log("You have logged in");

}

}

1. 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.

Ans = package instanceee;

import java.util.Scanner;

public class Employee {

int id;

String name;

double salary;

double netSalary;

double percentage;

double expense;

double newSalary;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public double getSalary() {

return salary;

}

public void setSalary(double salary) {

this.salary = salary;

}

public double applyRaise(double percentage) {

percentage += (this.salary\*2)/100 ;

salary += percentage;

System.***out***.println("Salary incremented is: "+percentage);

return percentage;

}

public double calculateTotalSalaryExpense(double expense) {

expense = percentage;

System.***out***.println("Salary incremented is: "+expense);

return expense;

}

public double updateSalary(double newSalary){

newSalary = salary+percentage;

setSalary(newSalary);

return newSalary;

}

public String toString() {

return "Employee [emp\_id = " + id + ", salary = " + salary + ", name = " + name+"]";

}

public static void main(String[] args) {

// **TODO** Auto-generated method stub

Scanner sc = new Scanner(System.***in***);

Employee emp = new Employee();

emp.setId(sc.nextInt());

emp.setName(sc.next());

sc.next();

emp.setSalary(sc.nextDouble());

emp.applyRaise(10222.23);

System.***out***.println(emp);

}

}