**ASSIGNMENT 5**

package com.cdac;

/\*1. Design and implement a class named InstanceCounter to track and

\* count the number of instances created from this class.\*/

public class InstanceCounter {

private static int *instantcounter*=0;

InstanceCounter()

{

*instantcounter*++;

}

public static int InstanceCounter() {

return *instantcounter*;

}

public static void main(String[] args) {

InstanceCounter ic1=new InstanceCounter();

InstanceCounter ic2=new InstanceCounter();

InstanceCounter ic3=new InstanceCounter();

System.*out*.println("the counter is:" +*instantcounter*);

}

}

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**

**package** com.cdac;

**import** java.util.ArrayList;

**import** java.util.Scanner;

**public** **class** Employee {

**private** **static** **int** *totalEmployees* ;

**private** **static** **double** *totalSalaryExpense* ;

**private** **int** id;

**private** String name;

**private** **double** salary;

**public** Employee(**int** id, String name, **double** salary) {

**this**.id = id;

**this**.name = name;

**this**.salary = salary;

*totalEmployees*++;

*totalSalaryExpense* += salary;

}

**public** **static** **int** getTotalEmployees() {

**return** *totalEmployees*;

}

**public** **static** **double** calculateTotalSalaryExpense() {

**return** *totalSalaryExpense*;

}

**public** **static** **void** applyRaise(ArrayList<Employee> employees, **double** percentage) {

**for** (Employee employee : employees) {

**double** raiseAmount = employee.salary \* percentage / 100;

employee.salary += raiseAmount;

*totalSalaryExpense* += raiseAmount;

}

}

**public** **void** updateSalary(**double** newSalary) {

*totalSalaryExpense* = *totalSalaryExpense* - **this**.salary + newSalary;

**this**.salary = newSalary;

}

**public** **int** getId() {

**return** id;

}

**public** String getName() {

**return** name;

}

**public** **double** getSalary() {

**return** salary;

}

@Override

**public** String toString() {

**return** "Employee [ID: " + id + ", Name: " + name + ", Salary: $" + salary + "]";

}

**public** **static** **void** main(String[] args) {

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

ArrayList<Employee> employees = **new** ArrayList<>();

**while** (**true**) {

System.***out***.println("\nEmployee Management System");

System.***out***.println("1. Add Employee");

System.***out***.println("2. View Total Employees");

System.***out***.println("3. Apply Raise to All Employees");

System.***out***.println("4. Update Employee Salary");

System.***out***.println("5. View Total Salary Expense");

System.***out***.println("6. Display All Employees");

System.***out***.println("7. Exit");

System.***out***.print("Enter your choice: ");

**int** choice = scanner.nextInt();

**switch** (choice) {

**case** 1:

System.***out***.print("Enter Employee ID: ");

**int** id = scanner.nextInt();

scanner.nextLine();

System.***out***.print("Enter Employee Name: ");

String name = scanner.nextLine();

System.***out***.print("Enter Employee Salary: ");

**double** salary = scanner.nextDouble();

Employee employee = **new** Employee(id, name, salary);

employees.add(employee);

System.***out***.println("Employee added successfully.");

**break**;

**case** 2:

System.***out***.println("Total Employees: " + Employee.*getTotalEmployees*());

**break**;

**case** 3:

System.***out***.print("Enter percentage raise: ");

**double** percentage = scanner.nextDouble();

Employee.*applyRaise*(employees, percentage);

System.***out***.println("Raise applied successfully.");

**break**;

**case** 4:

System.***out***.print("Enter Employee ID to update salary: ");

**int** updateId = scanner.nextInt();

**boolean** found = **false**;

**for** (Employee emp : employees) {

**if** (emp.getId() == updateId) {

System.***out***.print("Enter new salary: ");

**double** newSalary = scanner.nextDouble();

emp.updateSalary(newSalary);

System.***out***.println("Salary updated successfully.");

found = **true**;

**break**;

}

}

**if** (!found) {

System.***out***.println("Employee not found.");

}

**break**;

**case** 5:

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

**break**;

**case** 6:

**for** (Employee emp : employees) {

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

}

**break**;

**case** 7:

System.***out***.println("Exiting...");

scanner.close();

**return**;

**default**:

System.***out***.println("Invalid choice. Please try again.");

}

}

}

}

**package** com.cdac;

/\*

\* 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 *loggerInstance* = **null**;

**private** StringBuilder logMessages;

**private** Logger() {

logMessages = **new** StringBuilder(); // Initialize logMessages

}

**public** **static** **synchronized** Logger getInstance() {

**if** (*loggerInstance* == **null**) {

*loggerInstance* = **new** Logger(); // Create the instance if it doesn't exist

}

**return** *loggerInstance*;

}

**public** **void** log(String message) {

logMessages.append(message).append("\n"); // Append the message to logMessages

}

**public** String getLog() {

**return** logMessages.toString();

}

**public** **void** clearLog() {

logMessages.setLength(0);

}

**public** **static** **void** main(String[] args) {

Logger logger = Logger.*getInstance*();

logger.log("Application started.");

logger.log("User logged in.");

logger.log("User performed an action.");

System.***out***.println("Log Messages:");

System.***out***.println(logger.getLog());

logger.clearLog();

System.***out***.println("Log after clearing:");

System.***out***.println(logger.getLog());

Logger logger2 = Logger.*getInstance*();

logger2.log("This is a new log entry after clearing.");

System.***out***.println("Log Messages from logger2:");

System.***out***.println(logger2.getLog());

}

}