**Note:**

1. This assignment is designed to practice static fields, static initializers, and static methods.
2. Understand the problem statement and use static and non-static wisely to solve the problem.
3. Use constructors, proper getter/setter methods, and toString() wherever required.

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

package in.cdac.Assignment5.Q1;

class Counter{

static int objcount;

public Counter(){

this.objcount= objcount;

objcount=objcount+1;

}

public static int getObjectCount() {

return objcount;

}

}

public class InstanceCounter {

public static void main(String[] args) {

Counter obj1 = new Counter(); // create first object

Counter obj2 = new Counter(); // create second object

Counter obj3 = new Counter(); // create third object

Counter obj4 = new Counter();

System.out.println("Number of objects created: " + Counter.getObjectCount()); // print current count of objects

}

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** in.cdac.Assignmet5.Q2;

**import** java.util.List;

**public** **class** Logger {

**private** **static** Logger *instance*;

**private** String logMessages;

**private** Logger() {

logMessages = "";

}

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

**if** (*instance* == **null**) {

*instance* = **new** Logger();

}

**return** *instance*;

}

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

logMessages += message + "\n";

}

**public** String getLog() {

**return** logMessages;

}

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

logMessages = "";

}

}

**package** in.cdac.Assignmet5.Q2;

**public** **class** programlog {

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

Logger logger = Logger.*getInstance*();

logger.log("Application started");

logger.log("User logged in");

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

logger.clearLog();

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

}

}

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.

**package** in.cdac.Assignment5.Q3;

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

**return** *totalEmployees*;

}

**public** **static** **void** applyRaise(**double** percentage) {

*totalSalaryExpense* \*= (1 + percentage / 100);

}

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

**return** *totalSalaryExpense*;

}

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

}

**public** String toString() {

**return** "Employee [id=" + id + ", name=" + name + ", salary=" + salary + "]";

}

}

package in.cdac.Assignment5.Q3;

import java.util.Scanner;

import java.util.Scanner;

public class program {

private static Employeeutil currentEmployee;

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

while (true) {

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

System.out.println("2. Apply Raise");

System.out.println("3. Calculate Total Salary Expense");

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

System.out.println("5. Print Employee");

System.out.println("6. Exit");

System.out.print("Choose an option: ");

int option = scanner.nextInt();

switch (option) {

case 1:

currentEmployee.getrecord();

break;

case 2:

currentEmployee.raisesalary();

break;

case 3:

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

break;

case 4:

System.out.print("Enter New Salary: ");

double newSalary = scanner.nextDouble();

double totalSalaryExpense -= salary;

double salary = newSalary;

double totalSalaryExpense += salary;

break;

case 5:

System.out.println(currentEmployee);

break;

case 6:

System.exit(0);

break;

default:

System.out.println("Invalid option");

}

}

}

}