Assignment no :5

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

Solu:

**package** org.example;

**public** **class** InstanceCounter {

**private** **static** **int** *instanceCount* = 0;

**public** InstanceCounter() {

*instanceCount*++;

}

**public** **static** **int** getInstanceCount() {

**return** *instanceCount*;

}

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

// Create instances of InstanceCounter

InstanceCounter obj1 = **new** InstanceCounter();

InstanceCounter obj2 = **new** InstanceCounter();

InstanceCounter obj3 = **new** InstanceCounter();

// Print the number of instances created

System.***out***.println("Number of instances created: " + InstanceCounter.*getInstanceCount*());

}

}

Output:-

Number of instances created: 3

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** program17;

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

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

**private** StringBuilder logMessages;

**private** Logger() {

logMessages = **new** StringBuilder();

}

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

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

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

}

**return** *instance*;

}

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

logMessages.append(message).append(System.*lineSeparator*());

}

**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("Processing data.");

logger.log("Application finished.");

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

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

logger.clearLog();

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

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

}

}

Output:

log Messages:

Application started.

Processing data.

Application finished.

Log Messages after clearing:

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.

Solution:**package** program18;

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

**private** String name;

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

**private** **static** **int** *totalEmployees* = 0;

**private** **static** **double** *totalEmployeessalary* = 0.0;

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

**this**.id = id;

**this**.name = name;

**this**.salary = salary;

}

**public** **static** **int** gettotalEmployees() {

**return** *totalEmployees*;

}

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

**if**(percentage < 0) {

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

**return**;

}

**double** raiseMultiplier = 1 + (percentage / 100);

**public** **static** **double** CalculatetotalsalaryExpense() {

**return** totalsalaryexpense;

}

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

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

**this**.salary = newSalary;

}

}

**public** String toString() {

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

}

}

Output:-

Total Employees: 2

Total Salary Expense: 110000.0

After salary update:

Employee ID: 1, Name: Alice, Salary: 55000.0

Total Salary Expense: 115000.0

Applying raise of 10% to all employees.

After applying 10% raise:

Employee ID: 1, Name: Alice, Salary: 55000.0

Employee ID: 2, Name: Bob, Salary: 60000.0

Total Salary Expense: 1150000