**Assignment no 5**

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

**Program code:**

**package CounterInstance;**

**public class InstanceCounter {**

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

**public InstanceCounter() {**

***instanceCount*++;**

**}**

**public static int getInstanceCount() {**

**return *instanceCount*;**

**}**

**@Override**

**public String toString() {**

**return "Instance of InstanceCounter created. Current count: " + *instanceCount*;**

**}**

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

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

**System.*out*.println(obj1);**

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

**System.*out*.println(obj2);**

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

**System.*out*.println(obj3);**

**// Print total number of instances created**

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

**}**

**}**

**Output:**

**Instance of InstanceCounter created. Current count: 1**

**Instance of InstanceCounter created. Current count: 2**

**Instance of InstanceCounter created. Current count: 3**

**Total 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.**

**Program code:**

**package CounterInstance;**

**class Logger {**

**private static Logger *instance* = null;**

**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("\n");**

**}**

**public String getLog() {**

**return logMessages.toString();**

**}**

**public void clearLog() {**

**logMessages.setLength(0);**

**}**

**}**

**public class Logclass {**

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

**Logger logger = Logger.*getInstance*();**

**logger.log("Application started.");**

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

**logger.log("User performed action A.");**

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

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

**logger.clearLog();**

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

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

**}**

**}**

**Output:**

**Log Messages:**

**Application started.**

**User logged in.**

**User performed action A.**

**Log after clearing:**

* **clearLog(): Clears all log messages.**

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.**

**Program code:**

**package CounterInstance;**

**import java.util.ArrayList;**

**import java.util.Scanner;**

**class Employee {**

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

**private static double *totalSalaryExpense* = 0.0;**

**private int id;**

**private String name;**

**private double salary;**

**static {**

***totalEmployees* = 0;**

***totalSalaryExpense* = 0.0;**

**}**

**{**

**id = ++*totalEmployees*;**

**}**

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

**this.name = name;**

**this.salary = salary;**

***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 emp : employees) {**

**double raiseAmount = emp.salary \* (percentage / 100);**

**emp.salary += raiseAmount;**

***totalSalaryExpense* += raiseAmount;**

**}**

**}**

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

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

**this.salary = newSalary;**

***totalSalaryExpense* += 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 class Main {**

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

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

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

**int choice;**

**do {**

**System.*out*.println("\n--- Employee Management Menu ---");**

**System.*out*.println("1. Add new employee");**

**System.*out*.println("2. View total number of employees");**

**System.*out*.println("3. View total salary expense");**

**System.*out*.println("4. Apply raise to all employees");**

**System.*out*.println("5. Update salary of an employee");**

**System.*out*.println("6. Display all employees");**

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

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

**choice = sc.nextInt();**

**sc.nextLine();**

**switch (choice) {**

**case 1:**

**System.*out*.print("Enter employee name: ");**

**String name = sc.nextLine();**

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

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

**Employee newEmployee = new Employee(name, salary);**

**employeeList.add(newEmployee);**

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

**break;**

**case 2:**

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

**break;**

**case 3:**

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

**break;**

**case 4:**

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

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

**Employee.*applyRaise*(employeeList, percentage);**

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

**break;**

**case 5:**

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

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

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

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

**boolean found = false;**

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

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

**emp.updateSalary(newSalary);**

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

**found = true;**

**break;**

**}**

**}**

**if (!found) {**

**System.*out*.println("Employee with ID " + id + " not found.");**

**}**

**break;**

**case 6:**

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

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

**}**

**break;**

**case 7:**

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

**break;**

**default:**

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

**break;**

**}**

**} while (choice != 7);**

**sc.close();**

**}**

**}**

**Output:** **--- Employee Management Menu ---**

**1. Add new employee**

**2. View total number of employees**

**3. View total salary expense**

**4. Apply raise to all employees**

**5. Update salary of an employee**

**6. Display all employees**

**7. Exit**

**Enter your choice: 1**

**Enter employee name: alice**

**Enter employee salary: 50000**

**Employee added successfully!**

**--- Employee Management Menu ---**

**1. Add new employee**

**2. View total number of employees**

**3. View total salary expense**

**4. Apply raise to all employees**

**5. Update salary of an employee**

**6. Display all employees**

**7. Exit**

**Enter your choice: 2**

**Total Employees: 1**

**--- Employee Management Menu ---**

**1. Add new employee**

**2. View total number of employees**

**3. View total salary expense**

**4. Apply raise to all employees**

**5. Update salary of an employee**

**6. Display all employees**

**7. Exit**

**Enter your choice: 6**

**Employee ID: 1, Name: alice, Salary: ₹50000.0**