

ASSIGNMENT NO.6

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

Ans: **package** InstanceCounter;

```
public class Question1 {  
    public static int count=0;  
    public void InstanceCounter() {  
        count+=1;  
        System.out.println("The value of count "+Question1.count);  
    }  
  
    public static void main(String[] args) {  
        Question1 q=new Question1();  
        q.InstanceCounter();  
        q.InstanceCounter();  
        Question1 q2=new Question1();  
        q2.InstanceCounter();  
        q2.InstanceCounter();  
    }  
}
```

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.

Ans: **package** logging_Message;

```
import java.util.Scanner;  
class Logging{  
    private String name;  
    private String message;  
    // private static int count=0;
```

ASSIGNMENT NO.6

```

Logging(){
    this(" ", " ");

}

Logging(String name,String message){
    this.name=name;
    this.message=message;

}

public void setname(String name) {
    this.name=name;
}

public void setmessage(String message) {
    this.message=message;
}

public String getname() {
    return this.name;
}

public String getmessage() {
    return this.message;
}

public String toString() {
    return "logging [Name=" + this.name + ", Message=" + this.message + "];"
}

public void clearLog() {
    this.message = "";
}

}

class Message{
private static Logging reference = null;
public static Logging getLog() {
    if(reference==null)
        reference=new Logging();
    return reference;
}

public void Checking(String Message) {
    Message+=reference.getmessage();
}

}

```

ASSIGNMENT NO.6

```

class Helper{
    Scanner sc=new Scanner(System.in);
    public void accept() {

        Logging m= Message.getLog();

        //

        System.out.println("Enter the meassge");
        m.setmessage(sc.next());
        // System.out.println("Enter the name");
        // m.setName(sc.next());

    }
    public void print() {

        Logging m= Message.getLog();

        System.out.println(" the meassge :"+m.getMessage());

        // System.out.println(" the name :"+m.getName());
        // System.out.println("tostring practise"+m.toString());

    }
    public void Log() {
        Logging m = Message.getLog();
        m.clearLog();
    }
    public static int menulist() {
        Scanner sc=new Scanner(System.in);

        System.out.println("0. Exit.");
        System.out.println("1. Accept Record.");
        System.out.println("2. Print Record.");
        System.out.print("Enter choice: ");
        int choice = sc.nextInt();
        return choice;

    }

}

public class Question2 {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Helper h =new Helper();
    }
}

```

ASSIGNMENT NO.6

```
int choice;

while((choice=Helper.menulist())!=0) {
    switch (choice) {
        case 1:
            h.accept();

            break;
        case 2:
            h.print();
            h.Log();
            System.out.println("Checking the value message ");
            h.accept();
            break;
    }
}

//      h.accept();
//      h.print();
//      h.clearLog();

//      h.Log();
//      h.print();
//

//      Helper h2 =new Helper();
//      h2.accept();
//
//      h2.print();

}
}
```

3. 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()`)

ASSIGNMENT NO.6

- 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** employee;

//question is not done correctly

```
class Employee{
    private static int totalemp=0;
    private String name;
    private int empid ;
    private double salary;

    Employee(){
        this( null,0 , 0.0);
    }

    Employee(String name,int empid,double salary){
//        Employee.totalemp=total;
        this.name=name;
        this.empid=empid;
        this.salary=salary;
    }

    public void employeeCount() {
        totalemp+=1;
        System.out.println("The value of count "+Employee.totalemp);
    }

    public static int getTotalemp() {
        return totalemp;
    }
    public void setTotalemp(int totalemp) {
        Employee.totalemp = totalemp;
    }
    public String getName() {
        return name;
    }
}
```

ASSIGNMENT NO.6

```
    public void setName(String name) {
        this.name = name;
    }
    public int getEmpid() {
        return empid;
    }
    public void setEmpid(int empid) {
        this.empid = empid;
    }
    public double getSalary() {
        return salary;
    }
    public void setSalary(double salary) {
        this.salary = salary;
    }
    @Override
    public String toString() {
        return "Employee [name=" + name + ", empid=" + empid + ", salary=" + salary
+ Employee.getTotalEmp()+ "]";
    }

}

public class question3e {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Employee emp1 = new Employee("prakash",1,6000);
        Employee emp2 = new Employee("ghan",528,56325);

        // emp1.toString();
        System.out.println(emp1);
        emp2.toString();
        System.out.println(emp2.toString());

    }
}
```

