

**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.

```
package ass61;
class InstanceCounter {
    private static int count=0;
    public InstanceCounter(){
        InstanceCounter.count+=1;
    }
    public static int getCount() {
        return count;
    }
    public String toString() {
        return InstanceCounter.count+"";
    }
}
public class Q1 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        InstanceCounter i=new InstanceCounter();
        InstanceCounter i1=new InstanceCounter();
        System.out.println(InstanceCounter.getCount());
    }

}
```

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.

## ASSIGNMENT NO.6

```
package ass61;

class InstanceCounter{

    private String message;

    public static int count=0;

    private static InstanceCounter c;

    static {

        c=null;

    }

    public String getMessage() {

        return message;

    }

    public void setMessage(String message) {

        this.message += message;

    }

    public void clear() {

        this.message="";

    }

    private InstanceCounter() {

        this.message="";

    }

    public static InstanceCounter getInstance() {

        if(c==null) {

            count +=1;

            c=new InstanceCounter();

        }

    }

}
```

## ASSIGNMENT NO.6

```
}  
  
    return c;  
  
}  
  
public String toString() {  
  
    return this.message+" "+InstanceCounter.count;  
  
}  
  
}  
  
public class Q1 {  
  
    public static void main(String[] args) {  
  
        // TODO Auto-generated method stub  
  
        InstanceCounter i=InstanceCounter.getInstance();  
  
        InstanceCounter i1=InstanceCounter.getInstance();  
  
        InstanceCounter i2=InstanceCounter.getInstance();  
  
        i.setMessage("hi");  
  
        i.clear();  
  
        i.setMessage("hello");  
  
        System.out.println(InstanceCounter.count+" "+i.getMessage());  
  
    }  
  
}
```

## ASSIGNMENT NO.6

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()`)
- 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 ass5q3;
```

```
import java.util.Scanner;
```

```
class Employee{
    private String name;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getEmpid() {
        return empid;
    }
    public void setEmpid(int empid) {
        this.empid = empid;
    }
    public float getSalary() {
        return salary;
    }
    public void setSalary(float salary) {
        this.salary = salary;
    }
    private int empid;
    private float salary;
```

## ASSIGNMENT NO.6

```

public static int count=0;
public Employee() {
    count+=1;
}
public static void getTotoalEmployee() {
    // TODO Auto-generated method stub
    System.out.println("total employee :"+count);
}
public void applyRaise(double d) {
    // TODO Auto-generated method stub
    this.salary+=d;
    System.out.println(this.salary);
}
public static void calculateSalarayExpense(Employee[] em) {
    // TODO Auto-generated method stub
    float total=0.0f;
    for(Employee employee:em) {
        total=employee.getSalary();
    }
    System.out.println(total);
}
public void updateSalary(double salary) {
    // TODO Auto-generated method stub
    this.salary=(float)salary;
    System.out.println(this.salary);
}
@Override
public String toString() {
    return "Employee [name=" + name + ", empid=" + empid + ", salary=" + salary
+ "]\n";
}
}

class EmployeeUtil{
    private static Scanner sc=new Scanner(System.in);
    public void acceptRecord(Employee...emp) {
        for(Employee em:emp) {
            sc.nextLine();
            em.setName(sc.nextLine());
            em.setEmpid(sc.nextInt());
            em.setSalary(sc.nextFloat());
        }
    }
    public void printRecord(Employee...emp) {
        for(Employee em:emp) {

```

## ASSIGNMENT NO.6

```

    }

    public class Program {

        public static void main(String[] args) {
            Employee emp=new Employee();
            Employee emp1=new Employee();
            EmployeeUtil em=new EmployeeUtil();
            int choice;
            while((choice = EmployeeUtil.menuList())!=0) {
                switch(choice) {
                    case 1:em.acceptRecord(emp,emp1);
                        break;
                    case 2:em.printRecord(emp,emp1);
                        break;
                }
            }
        }
    }
}

```