Experiment No.2

Title: Implementations the concept of static keyword.
Aim: Understanding the concepts of static variable and static method.

Theory:

Java static keyword

The static keyword in Java is used for memory management mainly. We can apply static keyword with variables, methods, blocks and nested classes. The static keyword belongs to the class than an instance of the class.

The static can be:

1. Variable (also known as a class variable)
2. Method (also known as a class method)
3. Block
4. Nested class

1) Java static variable

If you declare any variable as static, it is known as a static variable.

- The static variable can be used to refer to the common property of all objects (which is not unique for each object), for example, the company name of employees, college name of students, etc.
- The static variable gets memory only once in the class area at the time of class loading.

Advantages of static variable

It makes your program **memory efficient** (i.e., it saves memory).

Understanding the problem without static variable

```
class Student{
    int rollno;
    String name;
    String college="ITS";
}
```

Suppose there are 500 students in my college, now all instance data members will get memory each time when the object is created. All students have its unique rollno and name, so instance data member is good in such case. Here, "college" refers to the common property of all objects. If we make it static, this field will get the memory only once.

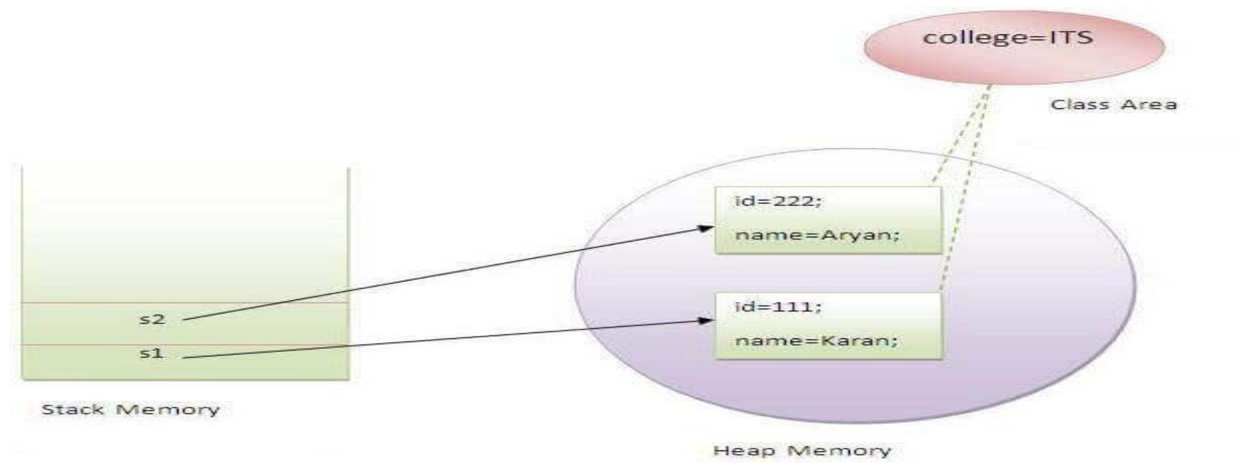
Java static property is shared to all objects.

Example of static variable

```
//Java Program to demonstrate the use of static variable
class Student{
   int rollno;//instance variable
   String name;
   static String college ="ITS";//static variable
   //constructor
   Student(int r, String n){
   rollno = r;
   name = n;
   }
   //method to display the values
   void display (){System.out.println(rollno+" "+name+" "+college);}
}
//Test class to show the values of objects
public class TestStaticVariable1{
 public static void main(String args[]){
 Student s1 = new Student(111,"Karan");
 Student s2 = new Student(222,"Aryan");
 //we can change the college of all objects by the single line of code
 //Student.college="BBDIT";
 s1.display();
 s2.display();
 }
}
```

Output:

```
111 Karan ITS
222 Aryan ITS
```



2) Java static method

If you apply static keyword with any method, it is known as static method.

- A static method belongs to the class rather than the object of a class.
- A static method can be invoked without the need for creating an instance of a class.
- A static method can access static data member and can change the value of it.

Example of static method

```
//Java Program to demonstrate the use of a static method.
class Student{
    int rollno;
    String name;
    static String college = "ITS";
    //static method to change the value of static variable
    static void change(){
    college = "BBDIT";
    }
    //constructor to initialize the variable
    Student(int r, String n){
    rollno = r;
    name = n;
    }
    //method to display values
    void display(){System.out.println(rollno+" "+name+" "+college);}
}
//Test class to create and display the values of object
public class TestStaticMethod{
   public static void main(String args[]){
   Student.change();//calling change method
   //creating objects
   Student s1 = new Student(111,"Karan");
   Student s2 = new Student(222,"Aryan");
   Student s3 = new Student(333,"Sonoo");
   //calling display method
   s1.display();
   s2.display();
   s3.display();
   }
}
```

Output:

```
111 Karan BBDIT
222 Aryan BBDIT
333 Sonoo BBDIT
```

Statement: Create class SavingsAccount. Use a static variable annualInterestRate to store the annual interest rate for all account holders. Each object of the class contains a private instance variable savingsBalance indicating the amount the saver currently has on deposit. Provide method calculateMonthlyInterest to calculate the monthly interest by multiplying the savingsBalance by annualInterestRate divided by 12this interest should be added to savingsBalance. Provide a static method modifyInterestRate that sets the annualInterestRate to a new value.
Write a program to test class SavingsAccount. Instantiate two savingsAccount objects, saver1 and saver2, with balances of Rs 2000.00 and Rs 3000.00, respectively. Set annualInterestRate to 4%, then calculate the monthly interest and print the new balances for both savers. Then set the annualInterestRate to 5%, calculate the next month's interest and print the new balances for both savers.

Program:

```
import java.util.*;

class SavingsAccount
{
        private static double annualInterestRate;
        private double savingsBalance;

        SavingsAccount()
        {
                savingsBalance=0;
                annualInterestRate=0;
        }

        SavingsAccount(double balance)
        {
                savingsBalance=balance;
                annualInterestRate=0;
```

```
          }

          void calculateMonthlyInterest()
          {
                    System.out.println("Current savings balance: "+savingsBalance);
                    double monthlyInterest;

                    monthlyInterest=(savingsBalance*annualInterestRate)/12;
                    savingsBalance+=monthlyInterest;

                    System.out.println("New savings balance: "+savingsBalance);

          }

          static void modifyInterestRate(double newInterestRate)
          {
                    annualInterestRate=newInterestRate;
          }
}

class Saving_test
{
          public static void main(String[] args)
          {
                    SavingsAccount saver1=new SavingsAccount(2000);
                    SavingsAccount saver2=new SavingsAccount(3000);

                    saver1.modifyInterestRate(.04);
                    saver1.calculateMonthlyInterest();

                    saver2.modifyInterestRate(.04);
                    saver2.calculateMonthlyInterest();

                    saver1.modifyInterestRate(.05);
                    saver1.calculateMonthlyInterest();

                    saver2.modifyInterestRate(.05);
                    saver2.calculateMonthlyInterest();
          }
}
```

Conclusion: Thus we implement the simple java program using static keyword.