static variable

1. The static variables are declared outside the method like instance variables. It is a variable which belongs to the class and not to object.
2. The static variable make use of the modifier “static ” before the data type.

   Eg:
   static int x;

   static String college=”Banasthali”;

3. The static variables are initialized only once, at the start of the execution. These variables will be initialized first, before the initialization of any instance variable.
4. For static variables there is only one copy of the variable is created irrespective of how many numbers of objects created in program, which shares across all the instances.
5. The static variable can be accessed directly by the class name and does not need any object.
6. The static variables are use-full for keeping track of global information such as number of objects instantiated from a class.
7. 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.

Understanding the problem without static variable

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

Suppose there are 50 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.

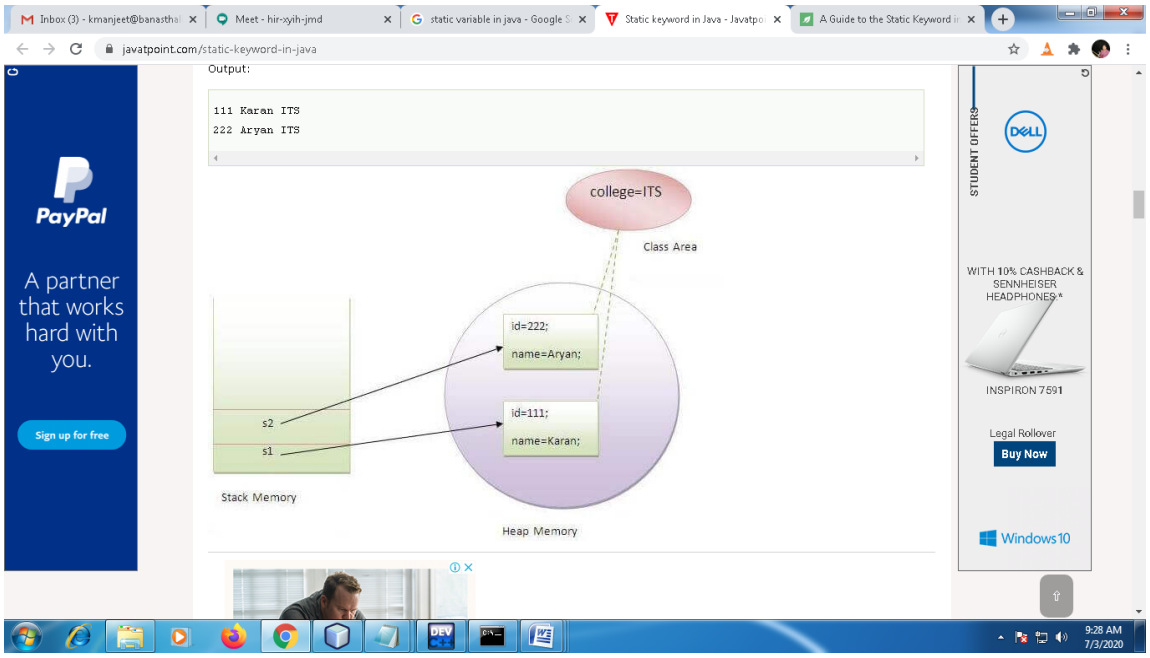
Example of static variable

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

```
 s2.display();

 }

}
```



Program of the counter without static variable

```
//Java Program to demonstrate the use of an instance variable

//which get memory each time when we create an object of the class.

class Counter{

int count=0;//will get memory each time when the instance is created

Counter(){

count++;//incrementing value

System.out.println(count);

}

public static void main(String args[]){

//Creating objects
```

```
Counter c1=new Counter();

Counter c2=new Counter();

Counter c3=new Counter();

}

}
```

Output:

```
1

1

1
```

Program of counter by static variable

```
//Java Program to illustrate the use of static variable which

//is shared with all objects.

class Counter{

static int count=0;//will get memory only once and retain its value


Counter(){

count++;//incrementing the value of static variable

System.out.println(count);

}


public static void main(String args[]){

//creating objects
```

Counter c1=new Counter();

Counter c2=new Counter();

Counter c3=new Counter();

}

}

Output:

1

2

3

Q: Write a Java program to create a class "STUDENT" with the data members Rollno, Name, Course, Branch, and Semester. Store them in an array of objects and display list of student. Roll no must be auto-increment and college name must be Banasthali.

Q: Write a Java program to create a class "STUDENT" with the data members Rollno, Name, Course, Branch, and Semester. Store them in an array of objects and display course,branch, semester and total fee of student course wise. Roll no must be auto-increment and college name must be Banasthali.

Static Method

- The static methods can be accessed like static variable through the class name without creating any instance or object.
- The static methods can only access static variables and methods.
- If you apply static keyword with any method, it is known as static method.

Syntax:

```
<static> <return_type> <method-name>(Parameter-list)
{
        //method body
}
```

Eg:

```
class Student2{
    int rollno;
    String name;
    static String college = "Banasthali";
    //static method to change the value of static variable
    static void change(){
    college = "BNS Vidyapith";
    }
    //method to initialize the variable
    void setValue(int r, String n){
    rollno = r;
    name = n;
    }
    //method to display values
    void display(){System.out.println(rollno+" "+name+" "+college);}

     public static void main(String args[]){
   // Student2.change();//calling change method

    //creating objects
    Student2 s1 = new Student2();
```

```
    Student2 s2 = new Student2();
    Student2 s3 = new Student2();

    //calling setValue method
    s1.setValue(1001, "Amit");
    s2.setValue(1002, "Rajesh");
    s3.setValue(1003, "Radha");
   //calling display method
    s1.display();
    s2.display();
    s3.display();
    }
}
```

Eg:

//Write a Java Program to get the cube of a given number using the static method

```
class Calculate{
  static int cube(int x){
  return x*x*x;
  }

  public static void main(String args[]){
  int result=Calculate.cube(5);
  System.out.println(result);
  }
}
```