

What is class Variable OR Static Field ?

\* It is a class level variable.

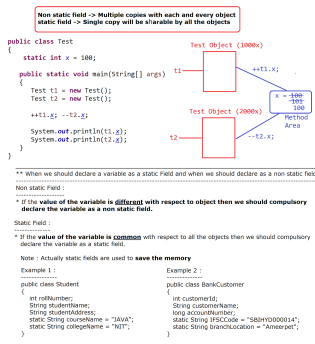
\* If we declare a variable inside the class with **static modifier** then it is called **static field**.

\* A static field is **automatically created and initialized** with default value at the time of **LOADING** the class file into JVM memory.

\* As far as its scope is concerned, it is available till the class is available in the JVM memory that means since the class will be un-loaded then only static variables value will be deleted from the special memory area called **Method Area**.

**Rule of Static field in Object creation :**

\* As we know static field can be directly accessible through class name, Object is not required.  
\* When we create an object in java then a **single copy of static field will be created and the same copy will be sharable by all the objects.**



WAP to show the uses of static and non static field :

```

package com.ravi.tf.ref;

public class Student
{
    int rollNumber;
    String studentName;
    String studentAddress;
    static String collegeName = "NIT";
    static String courseName = "Java";

    public void setStudentData(int roll, String name, String address)
    {
        rollNumber = roll;
        studentName = name;
        studentAddress = address;
    }

    public void getStudentData()
    {
        System.out.println("Student Roll is : "+rollNumber);
        System.out.println("Student Name is : "+studentName);
        System.out.println("Student Address is : "+studentAddress);
        System.out.println("College Name is : "+collegeName);
        System.out.println("Course Name is : "+courseName);
    }
}

package com.ravi.tf.ref;

public class StudentDemo
{
    public static void main(String[] args)
    {
        Student sunit = new Student();
        sunit.setStudentData(101, "Sunit", "Amrapur");
        sunit.getStudentData();

        System.out.println(".....");

        Student sunit1 = new Student();
        sunit1.setStudentData(102, "Sunit", "S R Nagar");
        sunit1.getStudentData();

        System.out.println(".....");

        Student sunit2 = new Student();
        sunit2.setStudentData(103, "Sunit", "Noida");
        sunit2.getStudentData();
    }
}

```

\*\*\*What is Data Hiding in Java ?

Data hiding is nothing but declaring our non static fields with private access modifier so our data will not be accessible from outer world that means no one can access our data directly from outside of the class.

\*We should provide the accessibility of our data through methods so we can perform **VALIDATION** ON DATA which are coming from outer world.

```

package com.ravi.data_hiding;

public class Customer
{
    private double balance = 10000; //Data Hiding

    public void deposit(double amount)
    {
        //Validation of outer world data
        if(amount <=0)
        {
            System.err.println("Amount can't be deposited");
        }
        else
        {
            balance = balance + amount;
            System.out.println("Balance after deposit is : "+balance);
        }
    }

    public void withdraw(double amount)
    {
        if(amount > balance)
        {
            System.err.println("Insufficient Balance!!!");
        }
        else
        {
            balance = balance - amount;
            System.out.println("Balance after withdraw is : "+balance);
        }
    }
}

package com.ravi.data_hiding;

public class BankApplication
{
    public static void main(String[] args)
    {
        Customer cunit = new Customer();
        cunit.deposit(10000);
        cunit.withdraw(10000);
    }
}

```