What is a instance variable OR non static field ?
----
* It is a variable which we are declaring at **class level.**

* If a non static variable is declared **inside the class but outside of the method** then it is called instance Variable OR non static field.
Example :
```
public class Car
{
   String carName;  //Instance Variable
}
```

* The life of non static variable will start at the time of creating the Object that means without an object we cannot think about non static variable.

```
public class Sample
{
    int x = 100;
    public static void main(String arrs[])
    {
        System.out.println(x); //error
    }
}
```

* As far as its scope is concerned, It is accessible from **anywhere in the class** as well as the accessibility level depends upon the access modifier we have applied on non static variable.

Initializing the non static variable through method without paramter using Scanner class :

```
package com.ravi.oop;

import java.util.Scanner;

//BLC
public class Manager
{
   int managerId;
   String managerName;
   double managerSalary;

   public void setManagerData()
   {
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter Manager id :");
      managerId = Integer.parseInt(sc.nextLine());

      System.out.print("Enter Manager Name :");
      managerName = sc.nextLine();

      System.out.print("Enter Manager Salary :");
      managerSalary = sc.nextDouble();
   }

   public void getManagerData()
   {
      System.out.println("Manager Id is :"+managerId);
      System.out.println("Manager Name is :"+managerName);
      System.out.println("Manager Salary is :"+managerSalary);
   }

}
```

```
package com.ravi.oop;

//ELC
public class ManagerDemo
{
   public static void main(String[] args)
   {
      Manager scott = new Manager();
      scott.setManagerData();
      scott.getManagerData();
   }
}
```

Note : In the above program we have initialized our non static variable with method without any parameter (using Scanner class)

How to initialize the non static variable with **method parameter**.

```
package com.ravi.oop;

//BLC
public class Employee
{
   int employeeId;
   String employeeName;
   double employeeSalary;

   public void setEmployeeData(int id, String name, double salary)
   {
     employeeId = id;
     employeeName = name;
     employeeSalary = salary;
   }

   public void getEmployeeData()
   {
      System.out.println("Employee Id is :"+employeeId);
      System.out.println("Employee Name is :"+employeeName);
      System.out.println("Employee Salary is :"+employeeSalary);
   }

}
```

```
package com.ravi.oop;

public class EmployeeDemo
{
   public static void main(String[] args)
   {
      Employee alen = new Employee();
      alen.setEmployeeData(111, "Mr Alen", 90000);
      alen.getEmployeeData();
   }
}
```

CONCLUSION :
----
* Upto Here we have learned total 3 ways to initialize our non static variables :

a) By using Object Reference (scott.roll = 101)
b) By using method without Parameter (Using Scanner class)
c) By using method with method parameter (Park Story)

Constructor [Only introduction] :
----
* If the **name of the class** and **name of the method** both are exactly same and It does not contain any return type then it is called Constructor.
Example :
```
public class Student
{
   public Student()  //Constructor
   {
   }
}
```

* In java, Whenever we write a class and if we don't write any type of constructor then by default java compiler will automatically add one default no argument constructor in the class.

Case 1 [User has not supplied any type of constructor]

Test.java
```
public class Test
{
   //User has not provided
  //any type of constructor
}
```

javac

Test.class
```
public class Test
{
   public Test()
   {
     //Default no argument constructor added by javac
   }
}
```

Case 2 :

Demo.java
```
public class Demo
{
   public Demo()
   {
      //Constructor written by user
   }
}
```

javac

Demo.class
```
public class Demo
{
  public Demo()
  {
    //Constructor written by user
  }
}
```

Compiler will not perform any action here

* Every java class must have at-least one constructor either implicitly added by java compiler OR explicitly written by user.

* The **access modifier** of default no argument constructor (added by java compiler) will depend upon the access modifier of the class that means if the class is public, default constructor access modifier will also be public, On the other hand if the class is non public then default constructor access modifier will also be non public.

Select C:\Windows\System32\cmd.exe
```
Microsoft Windows [Version 10.0.19045.5737]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user\Desktop\Batch 46>javac Test.java

C:\Users\user\Desktop\Batch 46>javap Test.class
Compiled from "Test.java"
public class Test {
  public Test();
}

C:\Users\user\Desktop\Batch 46>javac Test.java

C:\Users\user\Desktop\Batch 46>javap Test.class
Compiled from "Test.java"
class Test {
  Test();
}

C:\Users\user\Desktop\Batch 46>
```

Test.java (Source Code)

```
public class Test
{

}
```