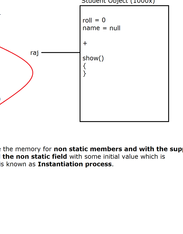
What is use of defult no argument constructor added by java compiler ?

```
public class Student
{
  int roll;
  String name;

  public Student() //default constructor
  {
  }
  public void show()
  {
    System.out.println(roll);
    System.out.println(name);
  }
}
public class StudentDemo
{
  public static void main(String [] args)
  {
    Student raj = new Student();
    raj.show();
  }
}
```



* new keyword is responsible to allocate the memory for **non static members and with the support of Java compiler it will initialize all the non static field** with some initial value which is known as default value. This process is known as **Instantiation process**.

Data types	:	Default value
byte		0
short		0
int		0
long		0
float		0.0
double		0.0
char		'\u0000' [Blank Space]
boolean		false
String		null
Object		null
Any reference variable		null

new keyword : Allocating the memory for non static member and initialized the non static field with default value.

* If a user will not supply any type of constructor then the default no argument constructor will help the programmer to create the object that means without the support default constructor object creation is not possible in java.

* Whenever we create an object in java with the help of **new keyword** then at-least one constructor must be invoked.

WAP that describes new keyword is responsible to initialize the non static field with default value.

```
package com.ravi.oop;
public class Student
{
    int roll;
    String name;

    public void show()
    {
        System.out.println("Roll Number is :"+roll);
        System.out.println("Name is :"+name);
    }
}

package com.ravi.oop;

public class StudentDemo {

    public static void main(String[] args)
    {
        Student raj = new Student();
        raj.show();
    }
}
```

WAP to initialize the non static field with parameter variable as per user requirement :

```
package com.ravi.oop;


//BLC
public class Employee
{
  int employeeNumber;
  String employeeName;
  double employeeSalary;
  char employeeGrade;

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

  public void calulateEmployeeGrade()
  {
     if(employeeSalary >= 75000)
     {
        employeeGrade = 'A';
     }
     else if(employeeSalary >= 50000)
     {
        employeeGrade = 'B';
     }
     else if(employeeSalary >= 25000)
     {
        employeeGrade = 'C';
     }
     else
     {
        employeeGrade = 'D';
     }
  }




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

}


package com.ravi.oop;


//ELC
public class EmployeeDemo
{
   public static void main(String[] args)
   {
      Employee smith = new Employee();
      smith.setEmployeeData(101, "Mr. Smith", 22000);
      smith.calulateEmployeeGrade();
      smith.getEmployeeData();

   }

}
```

Role of non static field while creating the Object :

---

* The life of the non static variable will start at the time of creating the object by using new keyword.
* Whenever we create an Object in java then a **separate copy** of all the non static fields are created with each and every object.

```
package com.ravi.oop;

public class Test
{
  int x = 90;

  public static void main(String[] args)
  {
     Test t1 = new Test();
     Test t2 = new Test();

       ++t1.x;   --t2.x;

     System.out.println(t1.x);
     System.out.println(t2.x);
  }
}
```