4-05-2020

Constructors in Java

6th week
033-332

02 February
Monday

Constructors are special type of method that is used to initialize the Object.

It has the same name as its class name.
Constructor must have no explicit return type.

Types of Constructors

Default Constructor
No parameters

```
class Demo{
    Demo()
    {
    }
}
```

Parameterized Constructor
It contains parameters (or arguments)

```
class Demo{
    Demo(int i, int j)
    {
    }
}
```

Various cases in java Constructors:

Case 1: When no default constructor is defined.
(Then java automatically provides a default constructor)

```
class Demo
{
    static int i;
    static String name;
    static float j;
    static double k;
    public static void main(String args[])
    {
```

Notes

```
Demo d = new Demo(); //default Constructor called
                     //here but we have not
                     //defined any constructor.
```

```
System.out.println( i);
System.out.println (name);
System.out.println(j);
System.out.println(k);
    }
}
```

Output would be:-

0
0.0
0.0
null

All output values are initializes by default Constructor provided by the compiler of JVM).

Case 2: When default constructor is provided by the user.

Example:-

```
class Demo{
    static int i;
    static String name;
    static float j;
    static double k;
Demo(){ i=200; name="SSC"; j=20; k=2.5f;
    }
public static void main(String args[])
{
```

```
Demo d = new Demo();
System.out.println(i);
System.out.println(name);
System.out.println(j);
System.out.println(k);
}
}
```

Now, Output will be;

200
SSC
20.0
2.5

Case 3: Parameterized Constructor

* Constructors with arguments is called parameterized Constructor.

```
class Demo
{
    static int i;
    static String name;
    static float j;
    static double k;
Demo(int x, String s, float f, double d)
{
    i=x;
    name=s;
    j=f;
    k=d;
}
```

Notes

```
public static void main(String args[])
{
Demo d = new Demo(5, "Ram", 4.7f, 16.5);
System.out.println("i = " + i);
System.out.println("name = " + name);
System.out.println("j = " + j);
System.out.println("k = " + k);
}
}
```

Now, output will be :-

i = 5
name = Ram
j = 4.7
k = 16.5

Case 4: Constructor Overloading

When more than one constructors are defined by any user then it is known as constructor overloading. It is called while initializing object of the class having similar arguments i.e; matching with arguments while creating object.

6th week
037-328

Example :-

```
class Demo
{
    static int roll;
    Static String name;
    Static float marks;
    Static double sports;

Demo(){roll=0; name="SSC";
       marks=55.7f; sports=2.5;
      }

Demo(int r, String nm, float m, double sp)
{
  roll=r; name=nm; marks=m; sports=sp;
}

public static void main(String args[])
{
    Demo d = new Demo();
System.out.println("roll="+roll);
System.out.println("Name=" +name);
System.out.println("Marks=" +marks);
System.out.println("Sports=" +sports);

Demo d1=new Demo(5, "Ram", 67.5f, 3.5);
System.out.println("Roll=" +roll);
System.out.println("Name=" +name);
System.out.println("Marks=" +marks);
System.out.println("Sports=" +sports);
}
}
```

Output will be :-

Roll = 0
Name = SSC
Marks = 55.7
Sports = 2.5
} default args. constructor worked.

Roll = 5
Name = Ram
Marks = 67.5
Sports = 3.5
} parameterized args. constructor worked.

In this way Constructor overloading works similar to function overloading. Now, one question is — if we have not defined default constructor but defined parameterized constructor only and we need to call default constructor while making object of the class then what will happen ?

Ans → Error ! We need to define both constructors parameterized and default if we have to call both of them. Here compiler does not provides us default constructor if we have defined parameterized one.

08 Sunday

Notes