

Constructors:

1. Object creation is not enough compulsory we should perform initialization then only the object is in a position to provide the response properly.
2. Whenever we are creating an object some piece of the code will be executed automatically to perform initialization of an object this piece of the code is nothing but constructor.
3. Hence the main objective of constructor is to perform initialization of an object.

Rules to write constructors:

1. Name of the constructor and name of the class must be same.
2. It is invoked by JVM automatically at time of object creation.
3. Return type concept is not applicable for constructor even void also by mistake if we are declaring the return type for the constructor we won't get any compile time error and runtime error compiler simply treats it as a method.

Example:

```
class Test
{
    void Test(){
        //it is not a constructor and it is a method
    }
}
```

3. It is legal (but should not be done) to have a method whose name is exactly same as class name.
4. The only applicable modifiers for the constructors are public, default, private, protected.
5. If we are using any other modifier we will get compile time error.

Types of Constructors:

There are three types of constructors :

1. No argument constructors
2. Parameterized constructors
3. Default constructors

1) No argument constructors

A constructor that doesn't accept any argument is called as no argument constructor.

```
package com.velocity.demo;

public class Test {

    int id;
    //program for no arg constructor
    Test() {
        id=10;
    }

    public static void main(String [] args) {
        Test test= new Test();
        System.out.println("Id value is "+test.id);
    }
}
```

Output-

Id value is 10

2) Default constructors :

A constructor that is automatically created by compiler is called as default constructor. Compiler will generate this constructor only if programmer doesn't write any constructor. If we write atleast one constructor then compiler will not generate default constructor.

```
package com.velocity.demo;

public class Test {

    int id;

    public static void main(String [] args) {
        Test test= new Test(); //Calling default constructor
    }
}
```

3) Parameterised Constructors:

The constructor with parameter is called as parameterized constructor.

```
package com.velocity.demo;

public class Test {

    int rollNum;
    String studentName;

    Test(int num, String name){
        rollNum=num;
        studentName=name;
    }

    public static void main(String [] args) {
        Test test= new Test(10, "Velocity"); //Calling
        parameterized constructor
        System.out.println("Student Roll Num is "+test.rollNum);
        System.out.println("Student Name is "+test.studentName);
    }
}
```

Output :

```
Student Roll Num is 10
Student Name is Velocity
```

Note:

- 1) Constructors doesn't have any return type
- 2) Constructors can be overloaded
- 3) For private constructors object creation is not possible
- 4) There are different ways to call constructors:
 - a) Test test = new Test();
 - b) new Test();
 - c) super();
 - d) this();