2.6 Constructor

❑ Providing Constructors for Your Classes

A class contains constructors that are invoked to create objects from the class blueprint. Constructor declarations look like method declarations—except that they use the name of the class and have no return type.

2.6.1 Constructor:

1. A constructor is a special method which initializes an object immediately upon creation.
2. It has the same name as class name in which it resides and it is syntactically similar to any method.
3. When a constructor is not defined, java executes a default constructor which initializes all numeric members to zero and other types to null or spaces.
4. Once defined, constructor is automatically called immediately after the object is created before new operator completes.
5. Constructors do not have return value, but they don't require "void" as implicit data type as data type of class constructor is the class type itself.

Parameterized constructor:

It is used to pass the values while creating the objects

Example:

```
class Rect
{
int length, breadth;
Rect(int l, int b) // parameterized constructor
{
length=l;
```

```
breadth=b;
}
public static void main(String args[])
{
Rect r = new Rect(4,5); //constructor with parameters
Rect r1 = new Rect(6,7);
System.out.println(“Area : ” +(r.length*r.breadth));
System.out.println(“Area : ” +(r1.length*r1.breadth));
}
}
```

```
public class Student{
public Student(){     // I ,  default constructor
System.out.println("Hello 1");
}
public Student(String name) {
 // II,  parameterized constructor with single parameter
System.out.println("Student name is " + name);
}

public Student(String name, int marks) {
// III, parameterized constructor with two parameters
System.out.println("Student name is " + name + " and marks are " +
marks);
}
public static void main(String args[]){
Student std1 = new Student();        // calls I
Student std2 = new Student("Mr.Reddy");       // calls II
Student std3 = new Student("Mr.Raju", 56);    // calls III
}
}
```