



SNS COLLEGE OF ENGINEERING



Kurumbapalayam(Po), Coimbatore - 641 107

Accredited by NAAC-UGC with 'A' Grade

Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of Information Technology

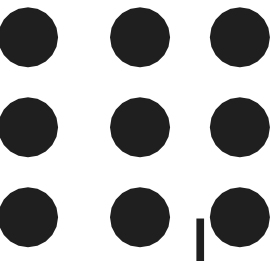
19CS204 OBJECT ORIENTED PROGRAMMING

I YEAR /II SEMESTER

Unit 2- BASIC FEATURES OF JAVA

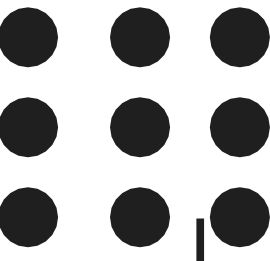
Constructors





Constructors

- It can be tedious to initialize all of the variables in a class each time an instance is created.
- Even when you add convenience methods, it would be simpler and more concise to have all of the setup done at the time the object is first created.
- Because the requirement for initialization is so common, Java allows objects to initialize themselves when they are created.
- This automatic initialization is performed through the use of a constructor



Constructors

- A constructor initializes an object immediately upon creation.
- It has the same name as the class in which it resides and is syntactically similar to a method.
- Once defined, the constructor is automatically called when the object is created, before the new operator completes.
- Constructors look a little strange because they have no return type, not even void



Constructors



Rules for Constructor

- Constructor name must be same as class name
- It cannot have a return type
- All classes have constructors by default
- If no constructors created for a class, java compiler creates it by default.

Types of constructor

- Default constructor (No arguments)
- Parameterized constructor (It has parameter and arguments can be passed to it).



Constructors



Default Constructor

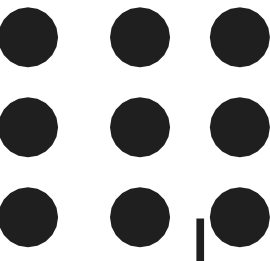
- A constructor that has no parameter is known as default constructor.
- If we don't define a constructor in a class, then compiler creates default constructor(with no arguments) for the class.
- Each time an object is created using new() keyword at least one constructor (it could be default constructor) is invoked to assign initial values to the data members of the same class.
- The default constructor automatically initializes all instance variables to their default values, which are zero, null, and false, for numeric types, reference types, and boolean, respectively.



Constructors

```
class student
{
    int mark1, mark2, mark3;
    String name;
    boolean boo;
    student()
    {
        System.out.println("Costructor Created");
        //mark1=90;
        mark2=90;
        mark3=70;
    }
    int total()
    {
        return mark1+mark2+mark3;
    }
}
```

```
public class mainstundet
{
    public static void main(String args[])
    {
        student s1=new student();
        int tot;
        tot=s1.total();
        System.out.println("Total is " +tot);
        System.out.println(s1.name);
        System.out.println("Default value "+s1.mark1);
        System.out.println("Default value for boolean is "+s1.boo);
    }
}
```



Constructors

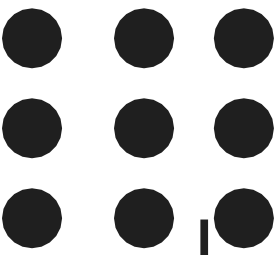
Parameterized constructor

Parameters are added to a constructor in the same way that they are added to a method, just declare them inside the parentheses after the constructor's name.

The parameterized constructor is used to provide different values to distinct objects. However, you can provide the same values also.

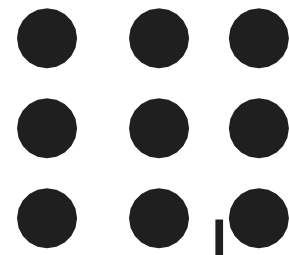


Constructors



```
class student
{
    int mark1, mark2, mark3;
    String name;
    student(int m1, int m2, int m3, String n)
    {
        System.out.println("Costructor Created");
        mark1=m1;
        mark2=m2;
        mark3=m3;
        name=n;
    }
    int total()
    {
        return mark1+mark2+mark3;
    }
}
```

```
public class mainstudent
{
    public static void main(String args[])
    {
        student s1=new student(90,90,70,"Ryan");
        int tot;
        tot=s1.total();
        System.out.println("Total is " +tot);
        System.out.println("Name is "+s1.name);
    }
}
```

THANK YOU