# Constructors

# Constructors

- A constructor is a special method that initializes values of instance variables
- It has the SAME NAME as that of the enclosing class
- It does not have a return type, not even void
- It is the first method that is called after an object is created (i.e. after memory allocation)
- Java supplies a default (parameterless) constructor if no constructor is defined in the class

# Constructors

```
class Square{
    private int side;

    public Square(){
        side = 1;
    }

    public void setSide(int x){
        if(x>-1 && x<11){
            side = x;
        }
        else{
            System.out.println("Input for side = "+x);
            System.out.println("Invalid: Value should be in the range 0-10");
        }
    }

    public int getSide(){
        return(side);
    }
}
```

- Adding a parameterless constructor to the Square class
- It initialises a new square object with side = 1

Example

# Java Source Code w/ Output

```
class Square{
    private int side;

    public Square(){
        side = 1;
    }

    public void setSide(int x){
        if(x>-1 && x<11){
            side = x;
        }
        else{
            System.out.println("Input for side = "+x);
            System.out.println("Invalid: Value should be in the range 0-10");
        }
    }

    public int getSide(){
        return(side);
    }
}

public class TestSquare{
    public static void main(String args[]){
        Square sq1 = new Square();
        Square sq2 = new Square();
        int s;

        s = sq1.getSide();
        System.out.println("\nOriginal side of sq1 = "+s);
        sq1.setSide(10);
        s = sq1.getSide();
        System.out.println("New side of sq1 = "+s);

        s = sq2.getSide();
        System.out.println("\nOriginal side of sq2 = "+s);
        sq2.setSide(7);
        s = sq2.getSide();
        System.out.println("New side of sq2 = "+s);
    }
}
```

```
javaprogs2020 — -bash — 61×9
TheLaptop:javaprogs2020 bertranddkhar$ javac TestSquare.java
TheLaptop:javaprogs2020 bertranddkhar$ java TestSquare

Original side of sq1 = 1
New side of sq1 = 10

Original side of sq2 = 1
New side of sq2 = 7
TheLaptop:javaprogs2020 bertranddkhar$
```

# Constructors

- A class may have multiple constructors, each one having a unique method signature i.e. accepts a different set of parameters in terms of number and type
- This is called constructor overloading
- A default (parameterless) constructor is not automatically added by Java if a class has at least one parameterized constructor i.e. it must be explicity defined by the programmer.

# Constructors

## Java Source Code w/ Output

TestSquare.java

```
class Square{
    private int side;

    public Square(){
        side = 1;
    }

    public Square(int s){
        side = s;
    }

    public void setSide(int x){
        if(x>-1 && x<11){
            side = x;
        }
        else{
            System.out.println("Input for side = "+x);
            System.out.println("Invalid: Value should be in the range 0-10");
        }
    }

    public int getSide(){
        return(side);
    }
}

public class TestSquare{
    public static void main(String args[]){
        Square sq1 = new Square();
        Square sq2 = new Square(6);
        int s;
```

javaprogs2020 — -bash — 61×10

```
TheLaptop:javaprogs2020 bertranddkhar$ javac TestSquare.java
TheLaptop:javaprogs2020 bertranddkhar$ java TestSquare

Original side of sq1 = 1
New side of sq1 = 10

Original side of sq2 = 6
New side of sq2 = 7
TheLaptop:javaprogs2020 bertranddkhar$
```

- Adding a parameterized constructor to the Square class
- It initialises a new square object with side as a user-defined value

# Copy Constructor

- There maybe a requirement to create an object that is identical to another object
- In such a situation, we can make use of a copy constructor
- It takes, as a parameter, an existing object reference variable
- Inside the copy constructor, we will copy piece-meal, the data members

```
public Square(Square sq){
        side = sq.getSide();
}
```

Creating objects from TestSquare:

Square sq1 = new Square(3);
Square sq2 = new **Square(sq1);**

# Java Source Code w/ Output

Example

TestSquare.java

```
class Square{
    private int side;

    public Square(){
        side = 1;
    }

    public Square(int s){
        side = s;
    }

    public Square(Square sq){
        side = sq.getSide();
    }

    public void setSide(int x){
        if(x>-1 && x<11){
            side = x;
        }
        else{
            System.out.println("Input for side = "+x);
            System.out.println("Invalid: Value should be in the range 0-10");
        }
    }

    public int getSide(){
        return(side);
    }
}

public class TestSquare{
    public static void main(String args[]){
        Square sq1 = new Square();        // Calls first constructor
        Square sq2 = new Square(6);       // Calls second constructor
        Square sq3 = new Square(sq2);     // Calls third constructor
        int s;
```

javaprogs2020 — -bash — 61×12

```
TheLaptop:javaprogs2020 bertranddkhar$ javac TestSquare.java
TheLaptop:javaprogs2020 bertranddkhar$ java TestSquare

Original side of sq1 = 1
New side of sq1 = 10

Original side of sq2 = 6
New side of sq2 = 7

Original side of sq3 = 6
New side of sq3 = 9
TheLaptop:javaprogs2020 bertranddkhar$
```