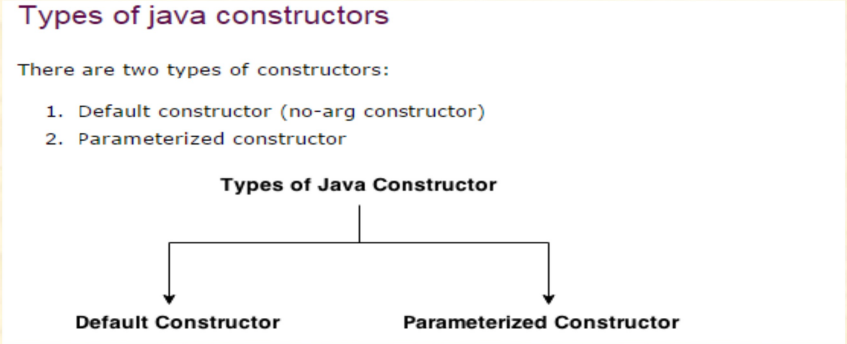
Constructor

Constructor

- Constructor is a special method that gets invoked “automatically” at the time of object creation.
- Constructor is normally used for initializing objects with default values unless different values are supplied.
- Constructor has the same name as the class name.
- Constructor cannot return values.
- A class can have more than one constructor as long as they have different signature (i.e., different input arguments syntax).



Defining a Constructor: Example

```
public class Counter {  int CounterIndex;

// Constructor
public Counter()
{
    CounterIndex = 0;
}
//Methods to update or access counter
public void increase()
{
    CounterIndex = CounterIndex + 1;
}
public void decrease()
{
    CounterIndex = CounterIndex - 1;
}
int getCounterIndex()
{
  return CounterIndex;
}
}
```


Trace counter value at each statement and What is the output ?

```
class MyClass {
  public static void main(String args[])
  {
      Counter counter1 = new Counter();  counter1.increase();
      int a = counter1.getCounterIndex();
      counter1.increase();
      int b = counter1.getCounterIndex();
      if ( a > b )
            counter1.increase();
      else
            counter1.decrease();

      System.out.println(counter1.getCounterIndex());
  }
}
```

Java parameterized constructor

A constructor that have parameters is known as parameterized constructor.

Why use parameterized constructor?

Parameterized constructor is used to provide different values to the distinct objects.

```
class Student4{
   int id;
   String name;

   Student4(int i,String n){
   id = i;
   name = n;
   }
   void display(){System.out.println(id+" "+name);}

   public static void main(String args[]){
   Student4 s1 = new Student4(111,"Karan");
   Student4 s2 = new Student4(222,"Aryan");
   s1.display();
   s2.display();
  }
}
```


Constructor Overloading

- Constructor overloading is a technique in Java in which a class can have any number of constructors that differ in parameter lists. The compiler differentiates these constructors by taking into account the number of parameters in the list and their type.

```
class Student5{
   int id;
   String name;
   int age;
   Student5(int i,String n){
   id = i;
   name = n;
   }
   Student5(int i,String n,int a){
   id = i;
   name = n;
   age=a;
   }
   void display(){System.out.println(id+" "+name+" "+age);}

   public static void main(String args[]){
   Student5 s1 = new Student5(111,"Karan");
   Student5 s2 = new Student5(222,"Aryan",25);
   s1.display();
   s2.display();
  }
}
```

Multiple Constructors

- Sometimes want to initialize in a number of different ways, depending on circumstance.
- This can be supported by having multiple constructors having different input arguments.

Multiple Constructors

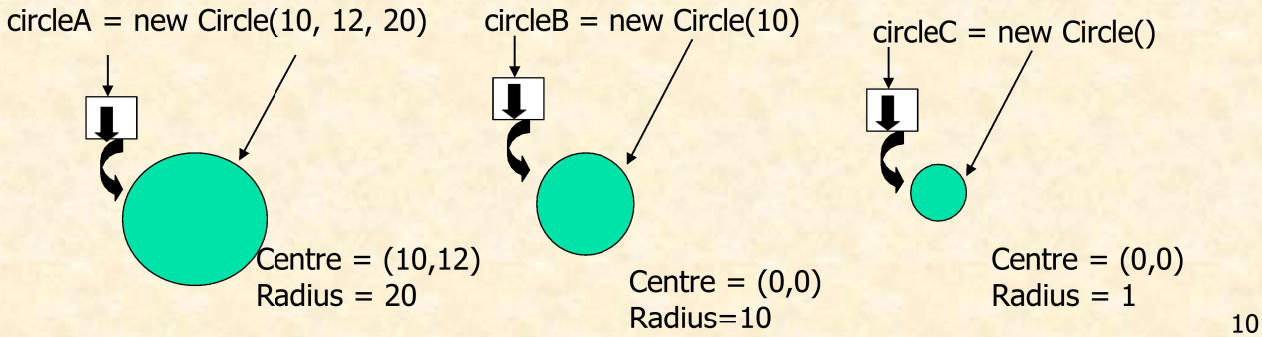
```
public class Circle {
    public double x,y,r; //instance variables
    // Constructors
    public Circle(double centreX, double cenreY, double radius) {
        x = centreX;  y = centreY; r = radius;
}
public Circle(double radius) { x=0; y=0; r = radius; }
public Circle() { x=0; y=0; r=1.0; }

 //Methods to return circumference and area
public double circumference() { return 2*3.14*r; }
public double area() { return 3.14 * r * r; }
}
```

Initializing with constructors

```
public class TestCircles {

        public static void main(String args[]){
            Circle circleA = new Circle( 10.0, 12.0, 20.0);  Circle circleB
            = new Circle(10.0);
            Circle circleC = new Circle();
        }
}
```



Java Copy Constructor

- There is no copy constructor in java. But, we can copy the values of one object to another like copy constructor in C++.

There are many ways to copy the values of one object into another in java. They are:

- By constructor
- By assigning the values of one object into another
- By clone() method of Object class

```
class Student6{
   int id;
   String name;
   Student6(int i,String n){
   id = i;
   name = n;
   }

   Student6(Student6 s){
   id = s.id;
   name =s.name;
   }
   void display(){System.out.println(id+" "+name);}

   public static void main(String args[]){
   Student6 s1 = new Student6(111,"Karan");
   Student6 s2 = new Student6(s1);
   s1.display();
   s2.display();
  }
}
```

Copying values without constructor

- We can copy the values of one object into another by assigning the objects values to another object. In this case, there is no need to create the constructor.

```
class Student7{
   int id;
   String name;
   Student7(int i,String n){
   id = i;
   name = n;
   }
   Student7(){}
   void display(){System.out.println(id+" "+name);}

   public static void main(String args[]){
   Student7 s1 = new Student7(111,"Karan");
   Student7 s2 = new Student7();
   s2.id=s1.id;
   s2.name=s1.name;
   s1.display();
   s2.display();
  }
}
```


Difference between constructor and method in java

Java Constructor	Java Method
Constructor is used to initialize the state of an object.	Method is used to expose behaviour of an object.
Constructor must not have return type.	Method must have return type.
Constructor is invoked implicitly.	Method is invoked explicitly.
The java compiler provides a default constructor if you don't have any constructor.	Method is not provided by compiler in any case.
Constructor name must be same as the class name.	Method name may or may not be same as class name.