

Constructors in Java - Detailed Notes

1. What is a Constructor?

- A constructor is a special method used to initialize objects.
- It has the same name as the class and no return type.
- Automatically called when an object is created.

2. Types of Constructors in Java:

A. Default Constructor:

- No parameters.
- Automatically provided by the compiler if no constructor is defined.

Example:

```
class MyClass {  
  
    MyClass() {  
  
        System.out.println("Default constructor");  
  
    }  
  
}
```

B. Parameterized Constructor:

- Accepts parameters to initialize values.

Example:

```
class MyClass {  
  
    int x;  
  
    MyClass(int value) {  
  
        x = value;  
  
    }  
  
}
```

```
}  
  
}
```

C. Copy Constructor (manual):

- Creates a new object using another object.

Example:

```
class MyClass {  
  
    int x;  
  
    MyClass(MyClass obj) {  
  
        x = obj.x;  
  
    }  
  
}
```

D. Private Constructor:

- Used in Singleton design pattern to restrict object creation.

Example:

```
class Singleton {  
  
    private Singleton() {}  
  
    static Singleton instance = new Singleton();  
  
}
```

3. Constructor Overloading:

- Multiple constructors in the same class with different parameter lists.

Example:

```
class MyClass {  
  
    MyClass() {}  
  
    MyClass(int x) {}  
  
}
```

```
MyClass(String name, int age) {}  
}
```

4. Keywords: this() and super()

- this(): Calls another constructor in the same class.
- super(): Calls a constructor of the parent class.

Example:

```
class Base {  
    Base(int a) {  
        System.out.println("Base constructor: " + a);  
    }  
}
```

```
class Derived extends Base {  
    Derived() {  
        super(10); // Calls Base(int a)  
        System.out.println("Derived constructor");  
    }  
}
```