PW SKILLS

JAVA With DSA & System Design

Assignment – Constructor in Java

Day - 21

---

1. **What is a Constructor?**
   Ans: Constructors initialize the state of an object during the time of object creation. The constructor is called when an object of a class is created. Constructors must have the same name as the class within which it is defined.

2. **What is Constructor Chaining?**
   Ans:
   - **Constructor chaining** is the process of calling one constructor from another constructor with respect to the current object.
   - One of the main uses of constructor chaining is to avoid duplicate codes while having multiple constructor (by means of constructor overloading) and make code more readable.
   - Constructor chaining can be done in two ways:
     - Within same class: It can be done using this() keyword for constructors in the same class
     - From base class: by using super() keyword to call the constructor from the base class.

3. **Can we call a subclass constructor from a superclass constructor?**
   Ans: No. We cannot call a subclass constructor from a superclass constructor. The reason being, that an instance of a subclass IS A instance of the superclass while the vice-a-versa is not true. Here, IS A is an OOPs concept where 'Foo IS A Bar' tells us that an instance of Foo can be held in a reference of Bar.

4. **What happens if you keep a return type for a constructor?**
   Ans:
   - We can overload a constructor so if we keep return type for a constructor it will be treated as a normal method.
   - Compiler gives a warning message that the method has a constructor name.

5. **What is No-argument constructor?**
   Ans: A constructor that has no parameter is known as the No-argument or Zero argument constructor. If we don't define a constructor in a class, then the compiler creates a constructor (with no arguments) for the class. And if we write a constructor with arguments or no arguments then the compiler does not create a default constructor.

6. **How is a No-argument constructor different from the default Constructor?**
   Ans:
   <u>Default Constructor in java:</u>
   - When we write a class without any constructor then at compilation time java compiler creates a default constructor in our class.
   - The accessibility modifier of the default constructor is the same as the accessibility modifier of class.
   - The allowed accessibility modifier is public and default.

     If our class have any constructor then java compiler does not create default constructor

No-argument Constructor in java:

- Default constructor added by java compiler this constructor does not have anything except super(); call.
- As a developer we can create our own constructor with no arguments is known as no-argument constructor.
- It can have all four accessibility modifiers as it is defined by the developer.
- So allowed accessibility modifiers are public, private, protected and default
- It can have logic including super call.

The common point between default and no-argument constructor:

- Both do not have any arguments.
- And one more point we need to remember that in no-argument constructor also by default the first statement will be a super () call which is added by the java compiler if it does not have.

7. When do we need Constructor Overloading?

Ans: Constructor overloading is possible in Java; it is called upon the parameters being executed. In general words, we can say it is a concept of having the various constructor with a different parameter list, in this way every constructor performs a different task.

We need to initialize the object in different ways as per the requirement for which we use Java constructor overloading. If we do not specify anything about a thread, then we can use the default but if we do want to specify.

8. What is Default Constructor Overloading? Explain with an example.

Ans:

Default Constructor Overloading:

The constructor that does not take any argument or all the arguments it takes are default variables and is referred to as the default constructor. It does not have function parameters. It is also known as a zero-argument constructor.

Example:

```
class Box
{
   double width, height, depth;

    // constructor used when all dimensions
   // specified
   Box(double w, double h, double d)
   {
      width = w;
      height = h;
      depth = d;
   }
    // constructor used when no dimensions
   // specified
   Box()
```

```
    {
        width = height = depth = 0;
    }

    // constructor used when cube is created
    Box(double len)
    {
        width = height = depth = len;
    }

    // compute and return volume
    double volume()
    {
        return width * height * depth;
    }
}

// Driver code
public class Test
{
    public static void main(String args[])
    {
        // create boxes using the various
        // constructors
        Box mybox1 = new Box(10, 20, 15);
        Box mybox2 = new Box();
        Box mycube = new Box(7);

        double vol;

        // get volume of first box
        vol = mybox1.volume();
        System.out.println(" Volume of mybox1 is " + vol);

        // get volume of second box
        vol = mybox2.volume();
        System.out.println(" Volume of mybox2 is " + vol);

        // get volume of cube
        vol = mycube.volume();
        System.out.println(" Volume of mycube is " + vol);
    }
}
```

Output:

```
Volume of mybox1 is 3000.0
Volume of mybox2 is 0.0
Volume of mycube is 343.0
```