Constructors

What is a Constructor?

A **constructor** is a special block of code (method-like) that is used to initialize objects of a class. It is automatically invoked at the time of object creation.

Key Points:

- **Constructor name and class name must be the same.**
- **Constructor can take parameters** (just like regular methods).
- **Constructors do not have a return type** (explicit return type declaration is **not allowed**, not even `void`).

Types of Constructors

1. Default Constructor

- If we **do not write** any constructor for a class, then the **compiler automatically generates** one for us. This is called the **default constructor**.
- The default constructor is **not visible in the code**.
- The compiler-generated default constructor is always a **0-argument constructor** with an **empty body**.

Example:

```
class Student {
    int id;
    String name;
}

public class Main {
    public static void main(String[] args) {
        Student s = new Student(); // Default constructor called
        System.out.println(s.id);  // Output: 0
        System.out.println(s.name); // Output: null
    }
}
```

2. User Defined Constructor

- If we **define at least one constructor** inside the class, then it is called a **user-defined constructor**.
- Main purpose: **Initialize instance variables** during object creation.
- Constructors can also contain other **initialization logic** needed at the time of object creation.
- You can define **multiple constructors** in a class using **constructor overloading** (different parameter lists).

Example:

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

    // User defined constructor
    Student(int i, String n) {
        id = i;
        name = n;
    }

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

public class Main {
    public static void main(String[] args) {
        Student s1 = new Student(101, "Alice");
        Student s2 = new Student(102, "Bob");
        s1.display(); // Output: 101 Alice
        s2.display(); // Output: 102 Bob
    }
}
```

Easy way to Remember :

Feature	Description
Name	Same as class name
Return type	Not allowed (not even void)

Parameters	Can have parameters
Default Constructor	Compiler-generated if no constructor is written
User Defined Constructor	Written by the programmer to initialize values
Multiple Constructors	Possible via overloading