



Chart Presentation



Constructors In Java

Constructors are special methods that are used to initialize the objects of the class. They are called when an object of the class is created.

Types of Constructors

- Default Constructor
- Parameterized Constructor

Example:

```

class Student {
    String name;
    int age;

    // Default Constructor
    Student() {
        name = "John";
        age = 20;
    }

    // Parameterized Constructor
    Student(String name, int age) {
        this.name = name;
        this.age = age;
    }
}
    
```

JAVA PROGRAM STRUCTURE

The structure of a Java program is as follows:

Example:

```

// Class Declaration
class Student {
    // Method Declaration
    void display() {
        // Method Body
    }
}

// Main Method Call
public class Main {
    public static void main(String[] args) {
        Student s = new Student();
        s.display();
    }
}
    
```

Access Modifiers

Access modifiers are used to restrict the access of the class members. They are:

- public
- private
- protected
- default (no modifier)

Example:

```

class Student {
    public String name;
    private int age;
    protected void display() {
        // Method Body
    }
}
    
```

JAVA STATEMENT

Java statements are used to perform operations. They are:

- Simple Statement
- Compound Statement
- Control Statement

Example:

```

// Simple Statement
int a = 10;

// Compound Statement
{
    // Multiple statements
}

// Control Statement
if (a > 0) {
    // Statement
}
    
```

INHERITANCE

Inheritance is a mechanism by which one class acquires the properties and behaviors of another class. It is used to create a hierarchy of classes.

Example:

```

class Animal {
    // Properties and Behaviors
}

class Dog extends Animal {
    // Properties and Behaviors
}
    
```

SYSTEMS ANALYSIS

Systems analysis is the process of identifying and defining the requirements for a system. It involves analyzing the current system and determining the needs for a new system.

Example:

```

class System {
    // Properties and Behaviors
}

class Analysis extends System {
    // Properties and Behaviors
}
    
```

Topic: Visibility In Digital Marketing

Visibility is the ability of a brand to be seen by its target audience. It is a key factor in digital marketing success.

Example:

```

class Visibility {
    // Properties and Behaviors
}

class DigitalMarketing extends Visibility {
    // Properties and Behaviors
}
    
```





