Constructor and this Keyword in Java

1. Constructor in Java

Definition:

A constructor in Java is a special method that is automatically called when an object of a class is created. It is used to initialize objects.

Types of Constructors:

1. Default Constructor
2. Parameterized Constructor
3. Copy Constructor (manually defined in Java)

Syntax:

```
class ClassName {
    // Constructor
    ClassName() {
        // initialization code
    }
}
```

Example:

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

    // Constructor
    Student() {
        id = 1;
        name = "Shalu";
    }

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

    public static void main(String[] args) {
        Student s = new Student(); // constructor called
        s.display();
    }
}
```

2. this Keyword in Java

Definition:

`this` is a reference variable in Java that refers to the current object.

Uses of `this` keyword:

1. To refer current class instance variables
2. To call current class methods or constructors
3. To pass the current object as an argument
4. To return the current object

Syntax:

```
this.variableName
this.methodName()
this()
```

Example 1: Referring to instance variables

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

    Student(int id, String name) {
        this.id = id;       // this refers to instance variable
        this.name = name;
    }

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

    public static void main(String[] args) {
        Student s = new Student(1, "Shalu");
        s.display();
    }
}
```

Example 2: Calling a constructor using `this()`

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

    Student() {
        this(2, "Viji");   // calling parameterized constructor
    }

    Student(int id, String name) {
        this.id = id;
```

```
        this.name = name;
    }

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

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