**this Keyword**

The this keyword in Java is a reference variable that refers to the current object. It is used in various contexts within a class to avoid ambiguity and to explicitly indicate that a member belongs to the current instance of the class.

Here are some common uses of the this keyword:

1. **Referring to Instance Variables**: When instance variables and method parameters have the same name, this is used to differentiate between them.

public class Example {

private int x;

public Example(int x) {

this.x = x; // 'this.x' refers to the instance variable, 'x' refers to the parameter

}

}

1. **Invoking Instance Methods**: this can be used to call another constructor in the same class (constructor chaining).

public class Example {

private int x;

private int y;

public Example(int x) {

this(x, 0); // Calls the constructor Example(int x, int y)

}

public Example(int x, int y) {

this.x = x;

this.y = y;

}

}

1. **Returning the Current Class Instance**: this can be returned to return the current class instance.

public class Example {

private int x;

public Example setX(int x) {

this.x = x;

return this; // Returning the current instance

}

public void printX() {

System.out.println(this.x);

}

public static void main(String[] args) {

Example example = new Example().setX(10);

example.printX(); // Output: 10

}

}

1. **Passing the Current Class Instance as a Parameter**: this can be used to pass the current class instance as a parameter to another method.

public class Example {

private int x;

public Example(int x) {

this.x = x;

}

public void display() {

System.out.println("Value of x: " + x);

}

public void show() {

this.display(); // 'this' refers to the current instance

}

public static void main(String[] args) {

Example example = new Example(10);

example.show(); // Output: Value of x: 10

}

}

1. **Calling Another Constructor in the Same Class**: When you have multiple constructors in a class, you can use this to call one constructor from another to avoid code duplication.

public class Example {

private int x;

private int y;

public Example() {

this(0, 0); // Calls the constructor Example(int x, int y)

}

public Example(int x, int y) {

this.x = x;

this.y = y;

}

}

**Example:** Here's an example that demonstrates multiple uses of the this keyword:

public class Person {

private String name;

private int age;

public Person() {

this("Unknown", 0); // Calls the constructor Person(String name, int age)

}

public Person(String name) {

this(name, 0); // Calls the constructor Person(String name, int age)

}

public Person(String name, int age) {

this.name = name;

this.age = age;

}

public void setName(String name) {

this.name = name; // 'this.name' refers to the instance variable, 'name' refers to the parameter

}

public void display() {

System.out.println("Name: " + this.name + ", Age: " + this.age);

}

public Person getPerson() {

return this; // Returning the current instance

}

public static void main(String[] args) {

Person person = new Person("John", 30);

person.display(); // Output: Name: John, Age: 30

person.setName("Jane");

person.display(); // Output: Name: Jane, Age: 30

Person newPerson = person.getPerson();

newPerson.display(); // Output: Name: Jane, Age: 30

}

}

In this example, this is used to:

1. Call another constructor in the same class.
2. Differentiate between instance variables and parameters with the same name.
3. Return the current instance of the class.