Yes, **method overloading** is possible in Java. Method overloading occurs when multiple methods in the same class have the **same name** but **different parameters**. These parameters can differ by:

1. The number of parameters
2. The type of parameters
3. The order of parameters (for example, swapping types)

Method overloading allows different versions of a method to be called based on the arguments passed to it, which can improve code readability and flexibility.

Example of Method Overloading

```java
class MathOperations {
    // Method to add two integers
    public int add(int a, int b) {
        return a + b;
    }

    // Overloaded method to add three integers
    public int add(int a, int b, int c) {
        return a + b + c;
    }

    // Overloaded method to add two doubles
    public double add(double a, double b) {
        return a + b;
    }
}

public class Main {
    public static void main(String[] args) {
        MathOperations math = new MathOperations();

        System.out.println("Sum of two integers: " + math.add(5, 10)); // Calls add(int, i
        System.out.println("Sum of three integers: " + math.add(5, 10, 15)); // Calls add(
        System.out.println("Sum of two doubles: " + math.add(5.5, 10.5)); // Calls add(dou
    }
}
```

Explanation

In the `MathOperations` class:

- There are three `add` methods with the same name but different parameters.
- The method `add(int a, int b)` is used to add two integers.
- The method `add(int a, int b, int c)` is overloaded to add three integers.
- The method `add(double a, double b)` is overloaded to add two `double` values.

Key Points

1. **Overloading is based on parameters**, not return type. You cannot overload a method by changing only its return type.
2. **Method signatures** must be different for overloaded methods. The method signature includes the method name and parameter types, so changes in the parameter list differentiate overloaded methods.
3. **Constructor Overloading** is also possible in Java, where multiple constructors with different parameters are defined within a class.

Advantages of Method Overloading

- **Improves Code Readability**: Using the same method name for similar operations makes the code more readable.
- **Enhances Flexibility**: Overloading provides the flexibility to call methods with different sets of parameters based on requirements.

Note

Method overloading is an example of **Compile-time Polymorphism** in Java, as the correct method is determined at compile time based on the parameters.