

# Java Method Overloading

Topics : [JAVA](#)

Written on [April 07, 2023](#)

In Java, method overloading allows a class to have multiple methods with the same name but different parameter lists. This means that a class can define multiple methods with the same name, as long as each method has a unique signature based on its parameter types.

When a method is overloaded, the Java compiler determines which version of the method to call based on the number, order, and types of the arguments passed to the method at runtime. The signature of a method includes the method name and the types of its parameters, but not the return type or any exceptions that the method may throw.

## Example :

```
int demoMethod(int x)
float demoMethod(float x)
double demoMethod(double x, double y)
```

Here is an example of method overloading:

```
public class Calculator {
    public int add(int x, int y) {
        return x + y;
    }

    public double add(double x, double y) {
        return x + y;
    }
}
```

In this example, the `Calculator` class defines two methods with the same name `add`, but with different parameter types. The first method takes two integers and returns an integer, while the second method takes two doubles and returns a double.

When the `add` method is called with integer arguments, the first version of the method is called. When the `add` method is called with double arguments, the second version of the method is called.

Method overloading is useful for providing multiple ways to perform the same operation, with different data types or numbers of parameters. It can make code more readable and easier to use, especially when different versions of a method have similar functionality.

### Example :

```
import java.util.*;

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

        int sum1 = m.add(33,44);
        System.out.println("The sum is: " + sum1);

        double sum2 = m.add(20.5,30.5);
        System.out.println("The sum is: " + sum2);

    }
    public int add(int x, int y) {
        return x + y;
    }

    public double add(double x, double y) {
        return x + y;
    }
}
```

### Output :

```
The sum is: 77
The sum is: 51.0
```