Method Overloading in Java

Method Overloading allows different methods to have the same name, but different signatures where the signature can differ by the number of input parameters or type of input parameters, or a mixture of both.

If a [class](https://www.javatpoint.com/object-and-class-in-java) has multiple methods having same name but different in parameters, it is known as **Method Overloading**.

Method overloading is also known as[***Compile-time Polymorphism***](https://www.geeksforgeeks.org/compile-time-polymorphism-in-java)***, Static Polymorphism, or***[***Early binding***](https://www.geeksforgeeks.org/difference-between-early-and-late-binding-in-java)in Java.

// Java program to demonstrate working of method

// overloading in Java

public class Sum {

    // Overloaded sum(). This sum takes two int parameters

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

    // Overloaded sum(). This sum takes three int parameters

    public int sum(int x, int y, int z)

    {

        return (x + y + z);

    }

    // Overloaded sum(). This sum takes two double

    // parameters

    public double sum(double x, double y)

    {

        return (x + y);

    }

    // Driver code

    public static void main(String args[])

    {

        Sum s = new Sum();

        System.out.println(s.sum(10, 20));

        System.out.println(s.sum(10, 20, 30));

        System.out.println(s.sum(10.5, 20.5));

    }

}

Different ways to achieve method overloading in java

* Changing the Number of Parameters.
* Changing Data Types of the Arguments.
* Changing the Order of the Parameters of Methods

## Advantages of Method Overloading

* Method overloading improves the Readability and reusability of the program.
* Method overloading reduces the complexity of the program.
* Using method overloading, programmers can perform a task efficiently and effectively.
* Using method overloading, it is possible to access methods performing related functions with slightly different arguments and types.
* Objects of a class can also be initialized in different ways using the constructors

**1) What is method overloading?**

When a class has more than one method with same name but different parameters, then we call those methods are overloaded. Overloaded methods will have same name but different number of arguments or different types of arguments.

**2) What is method signature? What are the things it consist of?**

In [Java](https://www.thoughtco.com/what-is-java-2034117), a method signature is part of the method declaration. It's the combination of the method name and the [parameter](https://www.thoughtco.com/parameter-2034268) list.

all the other elements of the method's declaration, such as modifiers, return type, parameter names, exception list, and body are not part of the signature.

**throw declaration cannot be part of the signature**.

public void print() {

    System.out.println("Signature is: print()");

}

public void print(int parameter) {

    System.out.println("Signature is: print(int)");

}

As we can see, the code compiles as the methods have different parameter type lists. In effect, the compiler can deterministically bind any call to one or the other.

Now let's test if it's legal to overload by adding the following method:

public int print() {

    System.out.println("Signature is: print()");

    return 0;

}

When we compile, we get a “method is already defined in class” error. That proves the method **return type is not part of the method signature**.

Same goes for modifiers also

**3) Can we declare one overloaded method as static and another one as non-static?**

Yes. Overloaded methods can be either static or non static.

**4) How do compiler differentiate overloaded methods from duplicate methods?**

Compiler uses method signature to check whether the method is overloaded or duplicated. Duplicate methods will have same method signatures i.e same name, same number of arguments and same types of arguments. Overloaded methods will also have same name but differ in number of arguments or else types of arguments

**5) Is it possible to have two methods in a class with same method signature but different return types?**

No, compiler will give duplicate method error. Compiler checks only method signature for duplication not the return types. If two methods have same method signature, straight away it gives compile time error.

**6) In “MyClass” , there is a method called “myMethod” with four different overloaded forms. All four different forms have different visibility ( private, protected, public and default). Is “myMethod” properly overloaded?**

Yes. Compiler checks only method signature for overloading of methods not the visibility of methods.

**8) Can we overload main() method?**

Yes, we can overload main() method. A class can have any number of main() methods but execution starts from **public static void main(String[] args)** only.

**9) Can we declare overloaded methods as final?**

Yes, we can declare overloaded methods as final.

**11) Overloading is the best example of dynamic binding. True or false?**

False. Overloading is the best example for static binding. (Click [here](https://javaconceptoftheday.com/difference-between-static-binding-and-dynamic-binding/) to see what is static binding and what is dynamic binding)

**12) Can overloaded method be overrided?**

Yes, we can override a method which is overloaded in super class.

Concept:

<https://www.geeksforgeeks.org/method-overloading-in-java/>

<https://www.javatpoint.com/method-overloading-in-java>

Interview Questions:

<https://javaconceptoftheday.com/important-java-interview-questions-on-method-overloading/>

<https://java2blog.com/method-overloading-and-overriding-interview-questions-in-java/#2_What_are_rules_of_method_overloading>

.