

What is method overloading in Java?

Method overloading in Java is an object-oriented programming concept that allows a programmer to declare two methods of the same name but with different method signatures, like a change in the argument list or a change in the type of argument. **Method overloading** is a powerful Java programming technique to declare a method that does a similar job but with a different kind of input. One of the most **popular examples of method overloading** is the `System.out.println()` method whose job is to print data on the console. This method is overloaded to accept all kinds of data types in Java.

You have the `println()` method which takes String, int, float, double, or even char in output. All of those methods are collectively referred to as an overloaded method in Java.

You should also know that **difference between method overloading and overriding** is also a popular Java interview question. In the next section, we will some important points about method overloading in Java and then a simple example of how to overload a method in Java.

By the way, if you are new to Java and object-oriented programming then I also recommend you to join a comprehensive Java course like **The Complete Java Masterclass on Udemy** to learn in a more structured way. This 80-hour long course is the most updated and comprehensive and you can get it for just \$10 on Udemy sales now.

Properties of method overloading in Java

1) *Overloaded methods* are bonded using **static binding in Java**. Which occurs during compile time i.e. when you compile a Java program. During the compilation process, the compiler bind method calls to the actual method.

2) Overloaded methods are fast because they are bonded during compile time and no check or binding is required during runtime.

3) Most important rule of method overloading in Java is that **two overloaded methods must have a different signature**.

Here is an example of *what does method signature means in Java*:

- A number of arguments to a method are part of the method signature.
- Type of argument to a method is also part of the method signature
- Order of argument also forms part of the method signature provided they are of a different type.
- The return type of method is not part of the method signature in Java.

These are some important points to remember while working in Java or going for Java interviews. If you need more object-oriented questions I suggest you check out **Grokking the Object-Oriented Design Interview** course from Educative.



Method Overloading Example in Java

Here is a list of methods and there is a corresponding overloaded method with the reason that How they are overloaded :

Original method :

```
public void show(String message){  
    System.out.println(message);  
}
```

Overloaded method: number of arguments is different

```
public void show(String message, boolean show){  
    System.out.println(message);  
}
```

Overloaded method: type of argument is different

```
public void show(Integer message){  
    System.out.println(message);  
}
```

Not an Overloaded method: the only return type is different

```
public boolean show(String message){  
    System.out.println(message);  
    return false;  
}
```

In the summary **method, overloading means multiple methods with the same name** but with a different signature. remember return type is not part of the method signature. method overloading is also completely different from method overriding which is a similar concept and we will see in the next article.