Method Overriding in Java:

1. If the child class implements the same method present in the parent class again, it is known as method overriding.

2. Method overriding helps us to classify a behavior that is specific to the child class.

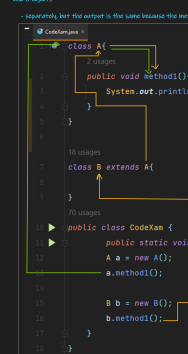
3. The subclass can override the method of the parent class only when the method is not declared as final.

4. Example :
- In the below code, we've created two classes: class A & class B.

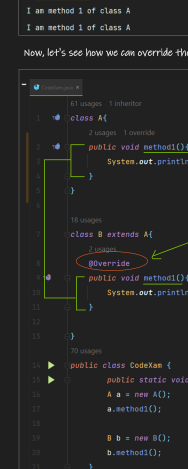
- class B is inheriting class A.

- In the main() method, we've created one object for both classes. We're running the method() method on class A and B objects

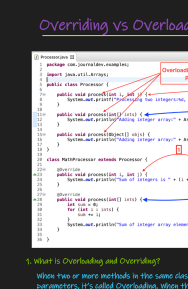
- separately, but the output is the same because the method() is defined in the parent class, i.e., class A.



Now, let's see how we can override the method() for class B :



Overriding vs Overloading in Java



1. What is Overloading and Overriding?

When two or more methods in the same class have the same name but different parameters, it's called Overloading. When the method signature (name and parameters) are the same in the superclass and the child class, it's called Overriding.

2. Overriding vs Overloading

Overriding implements Runtime Polymorphism whereas Overloading implements Compile time polymorphism.
The method Overriding occurs between superclass and subclass. Overloading occurs between the methods in the same class.
Overriding methods have the same signature i.e. same name and method arguments.
Overloaded method names are the same but the parameters are different.
With Overloading, the method to call is determined at the compile-time. With overriding, the method call is determined at the runtime based on the object type.
If overriding breaks, it can cause serious issues in our program because the effect will be visible at runtime. Whereas if overloading breaks, the compile-time error will come and it's easy to fix.

3. Overloading and Overriding Example

Here is an example of overloading and overriding in a Java program.