### **Method Overloading in Java**

Method Overloading is a **Compile time polymorphism**. In method overloading, more than one method shares the same method name with a different signature in the class. In method overloading, the return type can or can not be the same, but we have to change the parameter because, in java, we can not achieve method overloading by changing only the return type of the method.

**static** **int** add(**int** a, **int** b) //Has 2 arguments/signatures

{

**return** a + b;

}

**static** **int** add(**int** a, **int** b, **int** c)//Has 3 arguments/signatures

{

**return** a + b + c;

}

T*hese are overloaded methods, differ by number of signatures.*

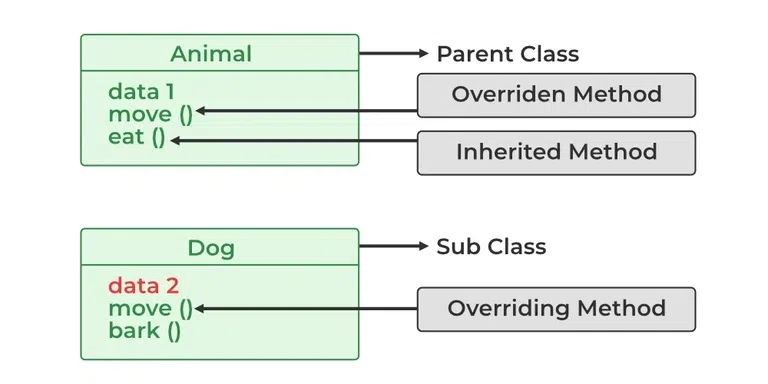
Method overloading can be done by 3 ways:

1. The number of parameters in two methods.
2. The data types of the parameters of methods.
3. The Order of the parameters of methods.

### **Method Overriding in Java**

Method Overriding is a **Run time polymorphism**. In method overriding, the derived class provides the specific implementation of the method that is already provided by the base class or parent class. In method overriding, the return type must be the same or co-variant (return type may vary in the same direction as the derived class).

Overriding is a feature that allows a subclass or child class to provide a specific implementation of a method that is already provided by one of its super-classes or parent classes. When a method in a subclass has the same name, the same parameters or signature, and the same return type(or sub-type) as a method in its super-class, then the method in the subclass is said to *override* the method in the super-class.



## **Rules for Java Method Overriding**

### **1. Overriding and Access Modifiers**

The access modifier for an overriding method can allow more, but not less, access than the overridden method. For example, a protected instance method in the superclass can be made public, but not private, in the subclass. Doing so will generate a compile-time error.

### **2. Final methods can not be overridden**

If we don’t want a method to be overridden, we declare it as final.

### **3. Static methods can not be overridden(Method Overriding vs Method Hiding):**

When you define a static method with the same signature as a static method in the base class, it is known as [method hiding](https://www.geeksforgeeks.org/can-we-overload-or-override-static-methods-in-java/). The following table summarizes what happens when you define a method with the same signature as a method in a super-class.

|  | **Superclass Instance Method** | **Superclass Static Method** |
| --- | --- | --- |
| **Subclass Instance Method** | Overrides | Generates a compile-time error |
| **Subclass Static Method** | Generates a compile-time error | Hides the subclass static method |

| **Method Overloading** | **Method Overriding** |
| --- | --- |
| Method overloading is a compile-time polymorphism. | Method overriding is a run-time polymorphism. |
| Method overloading helps to increase the readability of the program. | Method overriding is used to grant the specific implementation of the method which is already provided by its parent class or superclass. |
| It occurs within the class. | It is performed in two classes with inheritance relationships. |
| Method overloading may or may not require inheritance, as it can happen within a Class | Method overriding always needs inheritance, Parent and Sub Class |
| In method overloading, methods must have the **same name and different signatures.** | In method overriding, methods must have the **same name and same signature.** |
| In method overloading, the return type can or can not be the same, but we just have to change the parameter. | In method overriding, the return type must be the same or co-variant. |
| Static binding is being used for overloaded methods. | Dynamic binding is being used for overriding methods. |
| Private and final methods can be overloaded. | Private and final methods can’t be overridden. |
| The argument list should be different while doing method overloading. | The argument list should be the same in method overriding. |