

Method Overriding in Java

If subclass (child class) has the same method as declared in the parent class, it is known as **method overriding in java**.

In other words, If subclass provides the specific implementation of the method that has been provided by one of its parent class, it is known as method overriding.

Usage of Java Method Overriding

- Method overriding is used to provide specific implementation of a method that is already provided by its super class.
- Method overriding is used for runtime polymorphism

Rules for Java Method Overriding

1. method must have same name as in the parent class
2. method must have same parameter as in the parent class.
3. must be IS-A relationship (inheritance).

Understanding the problem without method overriding

Let's understand the problem that we may face in the program if we don't use method overriding.

```
class Vehicle{  
    void run(){System.out.println("Vehicle is running");}  
}  
class Bike extends Vehicle{  
  
    public static void main(String args[]){  
        Bike obj = new Bike();  
        obj.run();  
    }  
}
```

Test it Now

Output:Vehicle is running

Problem is that I have to provide a specific implementation of run() method in subclass that is why we use method overriding.

Example of method overriding

In this example, we have defined the run method in the subclass as defined in the parent class but it has some specific implementation. The name and parameter of the method is same and there is IS-A relationship between the classes, so there is method overriding.

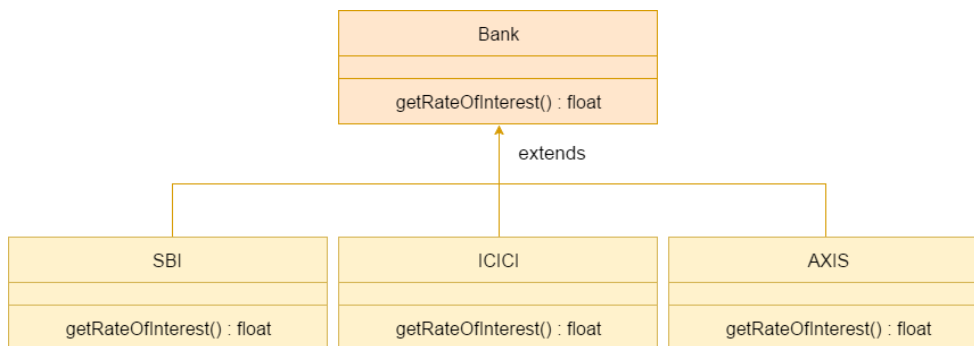
```
class Vehicle{  
    void run(){System.out.println("Vehicle is running");}  
}  
class Bike2 extends Vehicle{  
    void run(){System.out.println("Bike is running safely");}  
  
    public static void main(String args[]){  
        Bike2 obj = new Bike2();  
        obj.run();  
    }  
}
```

Test it Now

Output:Bike is running safely

Real example of Java Method Overriding

Consider a scenario, Bank is a class that provides functionality to get rate of interest. But, rate of interest varies according to banks. For example, SBI, ICICI and AXIS banks could provide 8%, 7% and 9% rate of interest.



```
class Bank{
    int getRateOfInterest(){return 0;}
}

class SBI extends Bank{
    int getRateOfInterest(){return 8;}
}

class ICICI extends Bank{
    int getRateOfInterest(){return 7;}
}

class AXIS extends Bank{
    int getRateOfInterest(){return 9;}
}

class Test2{
    public static void main(String args[]){
        SBI s=new SBI();
        ICICI i=new ICICI();
    }
}
```

```
AXIS a=new AXIS();  
System.out.println("SBI Rate of Interest: "+s.getRateOfInterest());  
System.out.println("ICICI Rate of Interest: "+i.getRateOfInterest());  
System.out.println("AXIS Rate of Interest: "+a.getRateOfInterest());  
}  
}
```

Test it Now

Output:

SBI Rate of Interest: 8

ICICI Rate of Interest: 7

AXIS Rate of Interest: 9

Can we override static method?

No, static method cannot be overridden. It can be proved by runtime polymorphism, so we will learn it later.

Why we cannot override static method?

because static method is bound with class whereas instance method is bound with object. Static belongs to class area and instance belongs to heap area.

Can we override java main method?

No, because main is a static method.

Difference between method Overloading and Method Overriding in java

[Click me for difference between method overloading and overriding](#)

More topics on Method Overriding (Not For Beginners)

Method Overriding with Access Modifier

Let's see the concept of method overriding with access modifier.

Exception Handling with Method Overriding

Let's see the concept of method overriding with exception handling.

[< prev](#)[next >](#)

Share this page



Latest 4 Tutorials

