|  |  |
| --- | --- |
| What is method overriding | If a subclass provides a specific implementation of a method that is already provided by its parent class, it is known as Method Overriding. It is used for runtime polymorphism and to implement the interface methods.   1. The method must have the same name as in the parent class. 2. The method must have the same signature as in the parent class. 3. Two classes must have an IS-A relationship between them. |
| Can we override the static method? | No, you can't override the static method because they are the part of the class, not the object. |
| Difference between method Overloading and Overriding. |  |
| Can we override the private methods? | No, we cannot override the private methods because the scope of private methods is limited to the class and we cannot access them outside of the class. |
| Can we change the scope of the overridden method in the subclass? | Yes, we can change the scope of the overridden method in the subclass. However, we must notice that we cannot decrease the accessibility of the method. The following point must be taken care of while changing the accessibility of the method.   1. The protected can be changed to public or default. 2. The default can be changed to public. 3. The public will always remain public. |
| Can we modify the throws clause of the superclass method while overriding it in the subclass? | Yes, we can modify the throws clause of the superclass method while overriding it in the subclass. However, there are some rules which are to be followed while overriding in case of exception handling.   1. If the superclass method does not declare an exception, subclass overridden method cannot declare the checked exception, but it can declare the unchecked exception. 2. If the superclass method declares an exception, subclass overridden method can declare same, subclass exception or no exception but cannot declare parent exception. |
| Output  Derived method called ... |  |
| Can we overload static method in Java? | Yes |
| Can we override static method in Java? | No |
| Base class method called with integer a = 10  The method() is overloaded in class Base whereas it is derived in class Derived with the double type as the parameter. In the method call, the integer is passed. |  |
| **What is covariant return type?** | Output: welcome to covariant return type |
| Method Hiding : in case of static method , static method can not override. |  |
|  |  |
| Can we change argument list of overridden method? | No |
| **Can we change return type of method in subclass while overriding?** | Yes co-variant method overriding in Java. |
| How do you call super class version of an overriding method in sub class? | You can call it using super keyword. For example if you have a method calculate() in both parent and child class then from child class you can invoke parent class method calculate() as super.calculate(). It's very similar to calling super class constructor from sub class as shown here. |
| Can we override constructor in Java? | No |
| Can we override final method in Java? | No |
| Can you overload or override main method in Java? | We can overload |
| **Can you achieve Runtime Polymorphism by data members?** | No, because method overriding is used to achieve runtime polymorphism and data members cannot be overridden. We can override the member functions but not the data members. Consider the example given below. |
| **What is the difference between static binding and dynamic binding?** | In case of the static binding, the type of the object is determined at compile-time whereas, in the dynamic binding, the type of the object is determined at runtime. |