Method overriding:

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, same parameters or signature, and 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.

If an object of a parent class is used to invoke the method, then the version in the parent class will be executed, but if an object of the subclass is used to invoke the method, then the version in the child class will be executed.

In this process, an overridden method is called through the reference variable of a superclass.

// A Simple Java program to demonstrate

// method overriding in java

// Base Class

**class** Parent {

**void** show()

    {

        System.out.println("Parent's show()");

    }

}

// Inherited class

**class** Child **extends** Parent {

    // This method overrides show() of Parent

    @Override

**void** show()

    {

        System.out.println("Child's show()");

    }

}

// Driver class

**class** Main {

**public** **static** **void** main(String[] args)

    {

        // If a Parent type reference refers

        // to a Parent object, then Parent's

        // show is called

        Parent obj1 = **new** Parent();

        obj1.show();

        // If a Parent type reference refers

        // to a Child object Child's show()

        // is called. This is called RUN TIME

        // POLYMORPHISM.

        Parent obj2 = **new** Child();

        obj2.show();

    }

}

**Final methods can not be overridden :**If we don’t want a method to be overridden, we declare it as [final](https://www.geeksforgeeks.org/final-keyword-java/).

**class** Parent {

    // Can't be overridden

**final** **void** show() {}

}

**class** Child **extends** Parent {

    // This would produce error

**void** show() {}

}

**Static methods can not be overridden:can not directly access instance varible and instance method.**

**Private methods can not be overridden :**