

Final Keyword

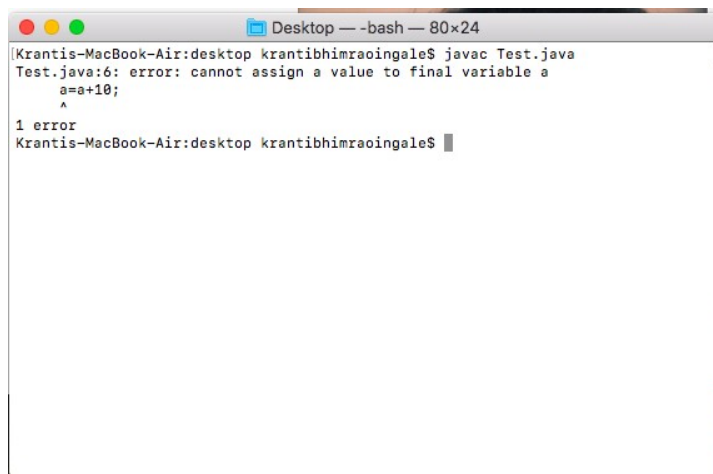
- 1) Final is the modifier applicable for **classes, methods and variables** (for all instance, Static and local variables).
- 2) The main advantage of final modifier is we can achieve security as no one can be allowed to change our implementation.
- 3) But the main disadvantage of final keyword is we are missing key benefits of OOPs like inheritance and polymorphism. Hence is there is no specific requirement never recommended to use final modifier.

Case 1: final keyword with Variables

If a variable is declared as final, then we cannot update value for that variable i.e., variable acts like constant.

```
class Test
{
    public static void main(String[] args)
    {
        final int a=10;
        a=a+10;
        System.out.println(a);
    }
}
```

Output:

A screenshot of a terminal window titled "Desktop -- -bash -- 80x24". The terminal shows the command "javac Test.java" being executed. The output is an error message: "Test.java:6: error: cannot assign a value to final variable a" followed by "a=a+10;" and a cursor. Below the error message, it says "1 error" and "Krantis-MacBook-Air:desktop krantibhimraoingale\$".

```
Desktop -- -bash -- 80x24
Krantis-MacBook-Air:desktop krantibhimraoingale$ javac Test.java
Test.java:6: error: cannot assign a value to final variable a
    a=a+10;
    ^
1 error
Krantis-MacBook-Air:desktop krantibhimraoingale$
```

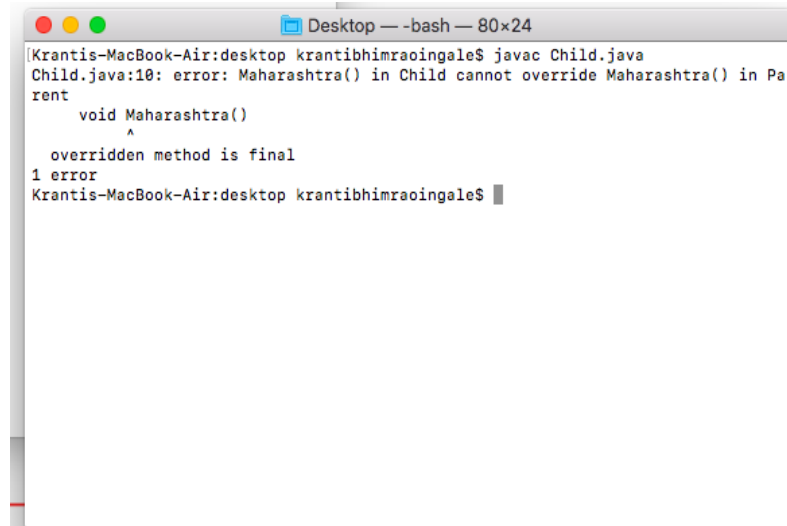
Case 2: final keyword with Methods

If a method is declared as final, then we cannot override that method into another that class i.e., method overriding not possible.

// overriding the method is not possible.

```
class Parent
{
    final void Maharashtra()
    {
        System.out.println("Odisha");
    }
}
class Child extends Parent
{
    void Maharashtra()
    {
        System.out.println("my state");
    }
    public static void main(String[] args)
    {
        Child c=new Child();
        c. Maharashtra ();
    }
}
```

output:

A screenshot of a terminal window titled "Desktop — -bash — 80x24". The terminal shows the command "javac Child.java" being executed. The output is an error message: "Child.java:10: error: Maharashtra() in Child cannot override Maharashtra() in Parent". Below the error message, the compiler shows the method signatures: "void Maharashtra()" for the parent class and "void Maharashtra()" for the child class, with a caret under the child's signature. The error message continues: "overridden method is final" and "1 error". The prompt "Krantis-MacBook-Air:desktop krantibhimraoingale\$" is shown at the bottom.

```
Desktop — -bash — 80x24
Krantis-MacBook-Air:desktop krantibhimraoingale$ javac Child.java
Child.java:10: error: Maharashtra() in Child cannot override Maharashtra() in Parent
    void Maharashtra()
        ^
    overridden method is final
1 error
Krantis-MacBook-Air:desktop krantibhimraoingale$
```

Case 3: final keyword with class

If a class is declared as final, then we cannot inherit that class i.e., we cannot create any child class for that final class.

```
final class Parent
{
}
class Child extends Parent
{
}
```

Output:

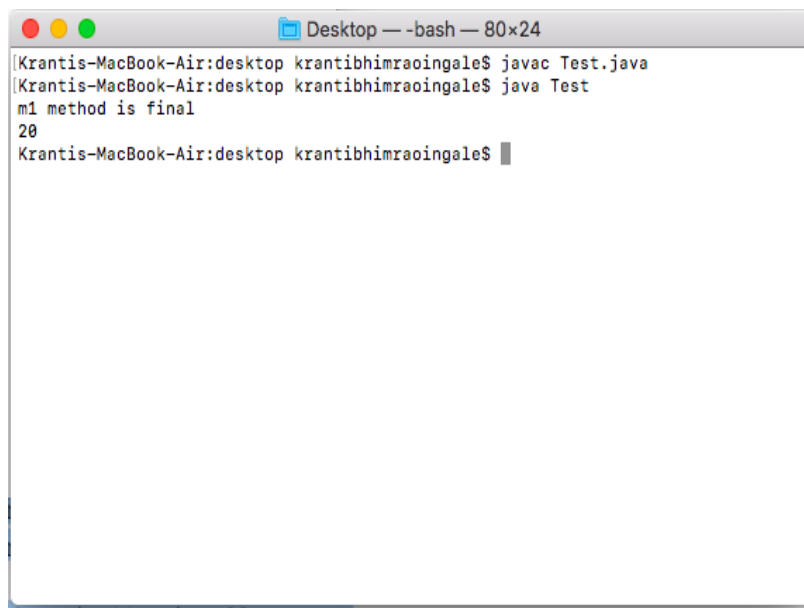
A screenshot of a macOS terminal window titled "Desktop — -bash — 80x24". The terminal shows the command "javac Child.java" being executed. The output is an error message: "Child.java:4: error: cannot inherit from final Parent", followed by the code snippet "class Child extends Parent" with a caret pointing to "Parent". Below the error message, it says "1 error" and then "Krantis-MacBook-Air:desktop krantibhimraoingale\$".

```
Desktop — -bash — 80x24
[Krantis-MacBook-Air:desktop krantibhimraoingale$ javac Child.java
Child.java:4: error: cannot inherit from final Parent
class Child extends Parent
                  ^
1 error
Krantis-MacBook-Air:desktop krantibhimraoingale$
```

Every method present inside a final class is always final by default but every variable present inside the final class need not be final.

```
final class Test
{
    int a=10;
    void m1()
    {
        System.out.println("m1 method is final");
        System.out.println(a+10);
    }
    public static void main(String[] args)
    {
        Test t=new Test();
        t.m1();
    }
}
```

Output:



```
Desktop — -bash — 80x24
[Krantis-MacBook-Air:desktop krantibhimraoingale$ javac Test.java
[Krantis-MacBook-Air:desktop krantibhimraoingale$ java Test
m1 method is final
20
Krantis-MacBook-Air:desktop krantibhimraoingale$
```