Java final keyword

In this tutorial, we will learn about Java final variables, methods and classes with examples.

In Java, the `final` keyword is used to denote constants. It can be used with variables, methods, and classes.

Once any entity (variable, method or class) is declared `final`, it can be assigned only once. That is,

- the final variable cannot be reinitialized with another value
- the final method cannot be overridden
- the final class cannot be extended

---

1. Java final Variable

In Java, we cannot change the value of a final variable. For example,

```
class Main {
  public static void main(String[] args) {

    // create a final variable
    final int AGE = 32;

    // try to change the final variable
    AGE = 45;
    System.out.println("Age: " + AGE);
  }
}
```

In the above program, we have created a final variable named age. And we have tried to change the value of the final variable.

When we run the program, we will get a compilation error with the following message.

```
cannot assign a value to final variable AGE
    AGE = 45;
    ^
```

Note: It is recommended to use uppercase to declare final variables in Java.

---

2. Java final Method

Before you learn about final methods and final classes, make sure you know about the Java Inheritance.

In Java, the `final` method cannot be overridden by the child class. For example,

```
class FinalDemo {
    // create a final method
    public final void display() {
      System.out.println("This is a final method.");
    }
}

class Main extends FinalDemo {
  // try to override final method
  public final void display() {
    System.out.println("The final method is overridden.");
  }

  public static void main(String[] args) {
    Main obj = new Main();
    obj.display();
  }
}
```


In the above example, we have created a final method named `display()` inside the `FinalDemo` class. Here, the Main class inherits the FinalDemo class.

We have tried to override the final method in the Main class. When we run the program, we will get a compilation error with the following message.

```
 display() in Main cannot override display() in FinalDemo
  public final void display() {
                    ^
  overridden method is final
```

---

3. Java final Class

In Java, the final class cannot be inherited by another class. For example,

```
// create a final class
final class FinalClass {
  public void display() {
    System.out.println("This is a final method.");
  }
}
```

```
// try to extend the final class
class Main extends FinalClass {
  public  void display() {
    System.out.println("The final method is overridden.");
  }

  public static void main(String[] args) {
    Main obj = new Main();
    obj.display();
  }
}
```

In the above example, we have created a final class named FinalClass. Here, we have tried to inherit the final class by the Main class.

When we run the program, we will get a compilation error with the following message.

```
cannot inherit from final FinalClass
class Main extends FinalClass {
                   ^
```