

Java Final Keyword

Final modifier is used to declare a field as final. It can be used with variable, method or a class.

If we declare a variable as final then it prevents its content from being modified. The variable acts like a constant. Final field must be initialized when it is declared.

If we declare a method as final then it prevents it from being overridden.

If we declare a class as final then it prevents from being inherited. We can't inherit final class in Java.

- **Example: Final variable**

In this example, we declared a final variable and later on trying to modify its value. But final variable can't be re-assigned so we get compile time error.

```
public class Test {  
    final int a = 10;  
    public static void main(String[] args) {  
        Test test = new Test();  
        test.a = 15;  
        System.out.println("a = "+test.a);  
    }  
}
```

- **Final Method**

A method which is declared using final keyword known as final method. It is useful when we want to prevent a method from overridden.

Example:

In this example, we are creating a final method learn() and trying to override it but due to final keyword compiler reports an error.

```

class Study
{
    final void learn()
    {
        System.out.println("learning something new");
    }
}

// concept of Inheritance
class Student extends Study
{
    void learn()
    {
        System.out.println("learning something interesting");
    }

    public static void main(String args[]) {
        Student object= new Student();
        object.learn();
    }
}

```

This will give a compile time error because the method is declared as final and thus, it cannot be overridden.

Let's take another example, where we will have a final variable and method as well.

```

class Cloth
{
    final int MAX_PRICE = 999; //final variable

```

```

final int MIN_PRICE = 699;

final void display()    //final method
{
    System.out.println("Maxprice is" + MAX_PRICE );
    System.out.println("Minprice is" + MIN_PRICE);
}
}

```

In the class above, the MAX_PRICE and MIN_PRICE variables are final hence their values cannot be changed once declared. Similarly the method display() is final which means even if some other class inherits the Cloth class, the definition of this method cannot be changed.

- **Final Class**

A class can also be declared as final. A class declared as final cannot be inherited.

We can create our own final class so that no other class can inherit it.

Example: Final Class

In this example, we created a final class ABC and trying to extend it from Demo class, but due to restrictions compiler reports an error.

```

final class ABC{
    int a = 10;
    void show() {
        System.out.println("a = "+a);
    }
}

class Demo extends ABC{
    public static void main(String[] args) {
        Demo demo = new Demo();
    }
}

```

```
    }  
}
```

- **Java Blank Final Variable**

Final variable that is not initialized at the time of declaration is called blank final variable. Java allows to declare a final variable without initialization but it should be initialized by the constructor only. It means we can set value for blank final variable in a constructor only.

Example:

In this example, we created a blank final variable and initialized it in a constructor which is acceptable.

```
class Demo  
{  
    final int a; // blank final variable  
    Demo()  
    {  
        a = 10; // initialized blank final  
    }  
    public static void main(String[] args)  
    {  
        Demo demo = new Demo();  
        System.out.println("a = "+demo.a);  
    }  
}
```

- **Static Blank Final Variable**

A blank final variable declared using static keyword is called static blank final variable. It can be initialized in static block only.

Static blank final variables are used to create static constant for the class.

Example

In this example, we are creating static blank final variable which is initialized within a static block. And see, we used class name to access that variable because for accessing static variable we don't need to create object of that class.

```
class Demo
{
    static final int a; //static blank final variable
    static
    {
        a = 10;        //initialized static blank final
    }
    public static void main(String[] args)
    {
        System.out.println("a = "+a);
    }
}
```