



A final variable in Java can be assigned a value only once, we can assign a value either in declaration or later.

```
final int i = 10;
i = 30; // Error because i is final.
```

A **blank final** variable in Java is a **final** variable that is not initialized during declaration. Below is a simple example of blank final.

```
// A simple blank final example
final int i;
i = 30;
```

### How are values assigned to blank final members of objects?

Values must be assigned in constructor.

```
// A sample Java program to demonstrate use and
// working of blank final
class Test
{
    // We can initialize here, but if we
    // initialize here, then all objects get
    // the same value. So we use blank final
    final int i;

    Test(int x)
    {
        // Since we have initialized above, we
        // must initialize i in constructor.
        // If we remove this line, we get compiler
        // error.
        i = x;
    }
}

// Driver Code
class Main
{
    public static void main(String args[])
    {
        Test t1 = new Test(10);
        System.out.println(t1.i);

        Test t2 = new Test(20);
        System.out.println(t2.i);
    }
}
```

[Run on IDE](#)

### Output:

```
10
20
```

*If we have more than one constructors or overloaded constructor in class, then blank final variable must be initialized in all of them. However constructor chaining can be used to initialize the blank final variable.*

```
// A Java program to demonstrate that we can
// use constructor chaining to initialize
// final members
class Test
{
    final public int i;

    Test(int val)    { this.i = val; }

    Test()
    {
        // Calling Test(int val)
        this(10);
    }

    public static void main(String[] args)
    {
        Test t1 = new Test();
        System.out.println(t1.i);

        Test t2 = new Test(20);
        System.out.println(t2.i);
    }
}
```

[Run on IDE](#)

### Output:

```
10
20
```

Blank final variables are used to create immutable objects (objects whose members can't be changed once initialized).