



Classes - Final members

And that's final!



Classes can have final members

The `final` keyword can be used in different contexts:

- Global constants - `public static final int SEVEN = 7;`
- Final instance fields - `private final int id = 2;`
- Final methods

```
public final void print() { /* ... */ }
```

- Final local variables - `final int result = a + b;`
 - Also parameters can be declared final
- Actually, the whole class can be declared final, but more on that later

Global constants

- The Java API is full of them:
<https://docs.oracle.com/javase/8/docs/api/constant-values.html>
- Cannot have their values changed after initialization
- Let's you use a name rather than remembering a value

```
public class PaymentTerms {
    public final static int NORMAL_CUSTOMER_PAYMENT_DAYS = 10;
    public final static int GOLD_CUSTOMER_PAYMENT_DAYS = 20;
    public final static int PLATINUM_CUSTOMER_PAYMENT_DAYS = 30;
    ...
}
```

Final instance variables

- Typically used when you want your class to be immutable
- More on immutability in a later lecture!
- Cannot have their values changed after initialization

Final methods

- Cannot be overridden using inheritance
- We'll talk more about this in a later lecture on inheritance

Final local variables

- Cannot be changed after initialization
- Signals to the reader of the code that this variable will not change further down
- Are considered a good idea by some programmers
- Parameters can also be final
 - Are like local variables
 - Should never change, so final is a good idea