Classes and Objects

Comparing Objects

- When comparing two objects using ==, the addresses (reference) of the objects are compared, rather than their content
- In Java, a boolean-valued instance method called `equals` is usually used to perform a comparison of two objects based on the contents of their fields.
- The `equals` method has one explicit parameter and it return `false` if the explicit parameter has the value `null`.

```
public boolean equals (Fraction other) {
   if (other != null && num == other.num && den == other.den) {
      return true;
   } else {
      return false;
   }
}
```

- The `equals` method can apply whatever criteria you choose to consider for equality.

Exercise

1. Modify the `equals` method such that two objects of `Fraction` are considered equal if the ratios of the `num` and `den` fields of the object are equal.

Displaying Objects

- When methods `print` and `println` are used to display objects, Java automatically calls an instance method `toString` (if provided) to convert the object to `String` value for printing.
- With the default `toString` method, `System.out.println(f)` outputs the the identifier of the class together with a memory reference to the object, e.g. `Fraction@1cc7c5`
- The `toString` can be overrided:

```
public String toString() {
     return num + "/" + den;
}
```

- Note that the `toString` method always return a String
- With the `toString` method above, `System.out.println(f)` would produce output of form `<num>/<den>`, e.g., `2/3`