Comparing and Displaying Objects

Recall:

- When creating an object, we can enter values into the object in several ways using assignment statements.

- We will once again, be referring to the class *fraction*

Example 1

fraction a = new fraction (3,4);

Will create a new object *a* of type object with the numerator initialized to 3 and the denominator initialized to 4.

Example 2

fraction a = new fraction (3,4);

fraction b = a;

Will copy the value from the reference *a* to the reference *b*. That is, *a* and *b* refer to the SAME object.

Example 3

fraction a = new fraction (3,4);

fraction b = new fraction (a);

Will create a new object *b* whose instance fields would have the same values as those of the *a*.

Comparing example 2 and 3

- In example 2, the expression a==b would have the value true.
  - i.e. both a and b refer to the same object.
- However in example 3, the expression a==b would have a value of false!
  - Although *a* and *b* have an identical num and den, they are not referring to the same object
  - i.e. a and b are distinct objects stored in different locations in memory

How do we compare two objects that do not refer to the same object?

Solution

- Compare the fields of the objects to each other.
- Better yet, write an instance method to do this!

- Build an instance method that will compare the fields of two objects of type *fraction*.

```
/* This instance method compares the fields of the implicit
   object with an object being passed in and returns a value
   of true if all the fields are identical*/
public boolean isequals (fraction x) {
     if ((x != null) && (num==x.num) && (den==x.den))
          return true;
     else
          return false;
}
```

Overview

- Example 1-3 reviewing the different ways to assign values to our objects
- Comparing objects
  - That refer to the same object
  - That do not refer to the same object
- Creating a method that will compare the fields of two objects
- Creating a method that will display our object nicely!