Announcements

• Apt set 2 due tomorrow!
• Jotto due February 5
  • Start early!

Last class

• Big-Oh
  • Recitation questions?
Today

• Comparing objects
  • Are they the same?
    • .equals()
  • What is the order?
    • compareTo()

Primitives

• int i = 5;
• int j = 5;

• if(i == j)
  • doSomething();
• else
  • doSomethingElse();
Objects

- int[] array1 = new int[5];
- int[] array2 = new int[5];
- if(array1 == array2)
  - doSomething();
- else
  - doSomethingElse();

**DO NOT USE == FOR OBJECTS!**
Objects

- Primitives are saved as values
- Objects are saved as reference values - value that points to another value somewhere in memory

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>5</td>
</tr>
<tr>
<td>s</td>
<td>@3971a0c2</td>
</tr>
<tr>
<td>@3971a0c2</td>
<td>“hello”</td>
</tr>
</tbody>
</table>

- int i = 5;
- String s = “hello”;

Pointers

MAN. I SUCK AT THIS GAME. CAN YOU GIVE ME A FEW POINTERS?

I HATE YOU.

0x3A28213A
0x6339392C,
0x7363682E.
How to compare objects?

• Is this the same teapot?

• teapot1.equals(teapot2);

.equals()

• Built in Java function for Object
• All objects inherit equals()
```java
ThreeInts a = new ThreeInts(5,5,5);
ThreeInts b = new ThreeInts(5,5,5);

if(a.equals(b))
    System.out.println("equal");
else
    System.out.println("not equal");
```

.equals() is a built-in Java function for Object. All objects inherit .equals(). You can override .equals() with your own code!

```java
Circle[] c = new Circle[numCircles];
for(int i = 0; i < numCircles; i++){
    c[i].setColor(colors[i%numColors]);
    i, colors[i%]
}
```
.equals()

```java
1 public boolean equals(Object obj){
2     if (obj == this) {
3         return true;
4     }
5     if (obj == null || obj.getClass() !=
          this.getClass()) {
6         return false;
7     }
8     YourObjectType temp = (YourObjectType) obj;
```
compareTo()

1 public class ThreeInts implements Comparable<ThreeInts>

2    public int compareTo(ThreeInts arg0) {
3       // code goes here
4       return 0;
5    }
Code time

- Create a class ComplexNumber
  - use ThreeInts as your guide
- ComplexNumber objects should have only two instance variables, myR and myI
- Write a .equals() and compareTo()
  - complex numbers should be compared using magnitudes

\[ \sqrt{r^2 + i^2} \]

Today

- Comparing objects
  - Are they the same?
    - .equals()
  - What is the order?
    - compareTo()