Computer Programming II Instructor: Greg Shaw

COP 3337

# The "==" Operator vs. the equals() Method

**Comparing Objects Using "=="**

* Java’s equality operator (==) returns **true** or **false** depending on whether its two operands are "equal"
* When comparing *primitive-type* variables for equality, as in

if (x == y)

**true** will be returned if the value stored in *x* is the same as the value stored in *y*; otherwise, **false** will be returned

* It works exactly the same way when the operands are *object* *variables*, but you must remember:
* The contents of an object variable is an object *reference* - the address of the object - and not the object itself
* So, if a and b are two *object* variables, the test

if (a == b)

returns **true** if and only if both *a* and *b* are pointing to the *same object*!

* If a and b are pointing to different objects, then "==" will always return **false**, even if the two objects "pointed to" contain the same values in their instance variables

**Behavior of the Inherited *equals()* Method**

* As defined in class **Object** (the ultimate superclass from which all other Java classes are derived), method *equals()* has exactly the same behavior as the equality operator (==)
* If you do not override *equals()* in your class (and your class is not derived from another class which *does* override it), then the *equals()* method of superclass **Object** will be called
* Program *NoOverriding.java* shows that the behavior of the inherited *equals()* method is the same as the "==" operator

**Overriding the *equals()* Method**

Recall that superclass methods are commonly *overridden* in subclasses to provide a more appropriate implementation

Method *equals()* is usually overridden to indicate whether the actual objects "pointed to" -- and not the pointers (or, *references*) -- are "equal" (i.e., contain the same *data*)

When overriding *equals()*, *we* get to say just what it means for two objects of our class to be "equal”

* Program *YesOverriding.java* shows how to override *equals()*