# The Object Class

Objects can ONLY be compared using .equals(), never using ==

- The Object class is the *superclass* of all other classes
- Classes such as Circle, and String, are *subclasses* of Object
- Subclasses *inherit*, or receive the methods of its superclass
- The object class includes methods for comparing objects (equals()) and representing an object as a string (toString())
- A subclass typically contains its own version of the equals() and toString() methods
    - This is done by redefining the superclass method (called *overriding*)

**Updating our Circle class:**

```
/**
 * Determines if the object is equal to another Circle object.
 * pre: c is a Circle object
 * post: true has been returned if the objects have
 * the same radii.  False has been returned otherwise
 */

public Boolean equals(Object c) {
      Circle testObj = (Circle)c;

      if (testObj.getRadius() == radius) {
            return(true);
      } else {
            return(false);
      }
}

/**
 * Returns a String that represents the Circle object.
 * pre: none
 * post: A String representing the Circle object has been returned.
 */

public String toString() {
      String circleString;

      circleString = "Circle has radius: " + radius;
      return(circleString);

}
```

The equals() method requires an Object parameter. As a result, it must be cast as the appropriate type

The concatenation operator + also invokes the toString() method of an object.

**Programming Exercises:**

a) Modify the Circle class to override the equals() and toString() methods, as shown in the previous section. Modify existing client code to test the new methods. (Submit this code)

b) Modify the Rectangle class to override the equals() and toString() methods. Two rectangles are equal when they both have the same length and width. Modify the existing client code to test the new method. (DO NOT submit this code yet)

c) Modify the Coin class to override the toString() method so that it indicates whether the coin is face up or face down. For example, "The coin is face up." Modify existing client code to test the new method. (Submit this code)

Add your code, including the client code, to the Google Doc: "ICS4U – Activity Submission Form".