Inheritance Con't

Abstract Class

- To be used when there are no good default methods in the superclass, and only the subclass programmer can know how to implement the method properly.

```
public abstract BankAccount{ // abstract class
    public abstract void withdraw(double amount); // method declaration
}
```

- It essentially provides a 'requirement' for subclasses to **have** to provide that method.
- Abstract methods have no implementation.
- It forces the subclass to implement the method.
- Cannot construct objects of abstract classes.
- **Concrete** classes are the opposite of abstract classes.
- There is a good example on JavaCoffeeBreak

THE COSMIC SUPERCLASS Object:

- In Java, every class that is defined without an explicit `extends` clause automatically extends the class Object.
- The class Object is the direct or indirect superclass of every class.
- Useful methods of the Object class:
  1. `String toString()`
  2. `boolean equals`

```
public class Tester {
    public static void main (String [] args){
        /**
         * Overrides Object's builtin toString method
         * @return the thing to display when object is called directly
         */
        public String toString(){
            return "Hello world";
        }
    }
}
```