Happy Monday!
Quiz

Define or explain how each of the following is used/useful:

• Classes
• Objects
• Interface
• Inheritance
Interfaces Practice

public interface Resizable {
    public void resize(double size);
    public boolean contains(int x, int y);
}

a) What methods would a class `Square` need? It should implement the `Resizable` interface

b) Finish the `Square` class
   1. Constructor should take ints x and y for `Location` and a `double` for size
   2. Draw rectangle centered on x,y using size. You can use:
      ```
      MyroLine line = new MyroLine(xstart, ystart, xend, yend);
      ```
   3. `contains()` should behave like you’d expect it to. `resize()` changes the size of the object by a factor (i.e., size=2 -> doubles the rectangle)
Using Interfaces

Resizable thingy;

thingy = new Square(...);

OR

thingy = new Circle(...);

thingy.resize(2);

Which method() gets called?
Caveat:

Resizable thingy;
Square mySquare;

thingy = mySquare;  // legal

mySquare = thingy;  // illegal!
• In addition to Square, supposed that Circle also implemented Resizable.

• In a different class, you have an array of 10 Resizable objects that has been initialized and are held in a class variable:
  
  ```java
  private Resizable[] myObjects;
  ```

Write a method

```java
public void enlarge(int x, int y)
```

The method takes x and y coordinates as parameters. Then, it finds all squares and circles in myObjects that overlap x,y and makes them shrink by 2.
Constants in Interfaces

• Constants are not re-declared in a class implementing the interface

```java
public interface IntExample {
    public static final int DAYS_IN_YEAR = 365;
    public static final Location ORIGIN = new Location(0, 0);
}
```
Using Constants

```java
public class Calendar implements IntExample {
    public static final int DAYS_IN_WEEK = 7;
    public static final int WEEKS_IN_YEAR = DAYS_IN_YEAR / DAYS_IN_WEEK;
}
```
Multiple Interfaces

What if we have two interfaces:

Movable:

```java
public interface Movable {
    public void move(double dx, double dy);
    public boolean contains(Location point);
}
```

Colorable:

```java
public interface Colorable {
    public void changeColor(Color aColor);
    public Color getColor();
}
```
Implementing Multiple Interfaces

- **HappyFace** must define all methods of its interfaces

```java
public class HappyFace implements Movable, Colorable {
    public void move(...) {
        ...
    }
    public void contains(...) {
        ...
    }
    public void changeColor(...) {
        ...
    }
    public Color getColor(...) {
        ...
    }
}
```

**Why useful?**
Summary of Interfaces

• Can be used as types of variables & parameters

• Include definitions of public constants and headers of public methods

• Classes
  – can implement one or more interfaces
  – must provide public method definitions for every method specified in interfaces