Model 1 Object Methods

In addition to providing constructors, getters, and setters, classes often provide `equals` and `toString` methods. These methods make it easier to work with objects of the class.

As a team, review the provided *Color.java* and *Point.java* files. Run each program to see how it works. Then answer the following questions using the source code (don't just guess).

Questions (15 min) Start time:

1. Based on the output of *Color.java*, what is the value of each expression below?

```
Color black = new Color();
Color other = new Color(0, 0, 0);
Color gold = new Color(255, 215, 0);
```

a) `black == other` false

b) `black == gold` false

c) `black.toString()` "#000000"

d) `black.equals(other)` true

e) `black.equals(gold)` false

f) `gold.toString()` "#ffd700"

2. What is the purpose of the `toString` method?

It returns a `String` representation of the `Color` (in HTML/CSS format). The `toString` method makes it easier to examine and debug objects.

3. Based on the output of *Point.java*, what is the value of each expression below?

```
Point p1 = new Point();
Point p2 = new Point(0, 0);
Point p3 = new Point(3, 3);
```

a) `p1 == p2` false

b) `p1.toString()` "(0, 0)"

c) `p3.toString()` "(3, 3)"

d) `p1.equals(p2)` true

e) `p1.equals("(0, 0)")` false

f) `p3.equals("(3, 3)")` false

4. What is the purpose of the `equals` method?

It determines whether two objects have the same attribute values. The `equals` method is useful for testing with `assertEquals`.

5. Examine *Point.java* again. What is the purpose of the `if`-statement in the `equals` method?

Since `equals` can take any type of `Object`, you need to check if the argument is a `Color` or `Point` instance before using it as such.

6. How could you modify the `equals` method to cause both #3e and #3f to return `true`?

Change the last line to `return this.toString().equals(obj);`

You could instead add `if (obj instanceof String)`, but since the `String.equals` method takes an `Object`, there's no need to convert the `obj` parameter before calling `String.equals`.