Javadoc & Testing

Before you're done writing a class:

— **Document:** User manual for users

— **Test:** Make sure it does everything your documentation says

Javadoc

Before each public element

```
/** Describe purpose of element
 *
 *  @tag Include all relevent tags
 */
public whatever ...
```

Javadoc Tags

Before `public class` include:

```
@author Ted Krovetz (1102)
@version 25 Jan 2016
```

Before each method include if relevant:

```
@throws ExceptionType explain when.
@param  paramName     explain purpose.
@return               what is meaning of return value.
```

Javadoc Process

— Write just enough to minimally explain the items.
— This class: learn the process.
— Include all relevant tags (@author, @version, etc).
— jGrasp: Bring file to front, click book icon.
— Minimize warnings when generating.
— Check HTML has every public item documented.

Testing

You must test your work thoroughly to trust it.

— Understand spec.

— Run code with:
    ~ simple inputs
    ~ extreme inputs (very big, very small, empty)
    ~ complex inputs
    ~ inputs you think might be hard for your code

— Try to "break" your code while staying within spec.

Two kinds of testing

— Test as you go:

~ Write small amount code
~ Write `main` with simple tests for it
~ Good for quick feedback

— Comprehensive test

~ Once your class is complete
~ Write big `main` that does wide range of tests
~ My test of your submissions is like this

Example Comprehensive Test: CSS Circle

```
public static void main(String[] args) {
    boolean pass = true;  // Assume Circle is good.
    Circle a = new Circle(1.0);
    Circle b = new Circle(2.0);
    pass = pass && eq(a.getRadius(), 1.0);
    pass = pass && eq(b.getRadius(), 2.0);
    pass = pass && eq(a.area(), Math.pow(1.0, 2.0) * Math.PI);
    pass = pass && eq(b.area(), Math.pow(2.0, 2.0) * Math.PI);
    pass = pass && eq(a.circumference(), 2.0 * 1.0 * Math.PI);
    pass = pass && eq(b.circumference(), 2.0 * 2.0 * Math.PI);
    System.out.println("Circle passes all tests: " + pass);
}
```

```
public static boolean eq(double a, double b) {
    return Math.abs(a-b) < 0.0001;
}
```

Why?

```
public static void main(String[] args) {
    double x = Math.sqrt(2.0);
    if (x * x == 2.0)
        System.out.println("equal");
    else
        System.out.println("not equal " + (x * x));
}
// Prints: not equal 2.0000000000000004
```

How to test exceptions

```
try {
    new Circle(-1.0);  // IllegalArgumentException expected
    pass = false;      // Should not have gotten here
}
catch (IllegalArgumentException e) {
    // This is wehere we expect to go, don't set pass to false
}
catch (Exception e) {
    pass = false      // Any other type of exception is incorrect
}
```