Testing

* Testing is the process by which we make sure that our code performs reliably and correctly.
* Nothing can be ‘proven’ to not be broken with testing
  + This is not math where you can write algorithms or proofs
  + As far as our tests have failed to break the code we believe it works correctly
  + Testing can never be truly 100% complete
* Functional testing
  + Testing to make sure something works correctly
    - You do not care if it is efficient
* Performance
  + Testing the efficiency and speed of an application/ piece of code

TDD (Test Driven Development)

* In TDD the development paradigm is to create the tests first. Then create the code/implementation.
  1. Write your interface (You get your method signatures).
  2. Write your tests (You know the behavior and expected outputs of your methods).
  3. Write an implementation of that interface and have it run against those tests.
* Why write tests first?
  1. Helps you diagnose edge cases
     + Edge cases are bizarre inputs
     + Very massive or unusual inputs
  2. Helps avoid method/code blind spots
     + Inputs your method should handle but do not
  3. Quickly diagnose where errors arise
     + Incredibly useful and time saving when you have a method that calls another method and another method and so on

Unit Test

* Unit test tests a single atomic piece of code
  + Atomic means that this test tests a single method that DOES NOT call on another method
* Type of functional test
* Type white box test
  + White box test means that you can see the code you are testing
* Positive tests
  + Test that something works as it should
  + Given the correct inputs you get the correct outputs
* Negative tests
  + Test that something fails as it should
  + Given incorrect inputs you get an appropriate error or failure

Example login testing

* Positive tests
  + Correct username and password logs me in =)
    - Ex Mark55 , markmanpass
    - Ex KimCool , kimpossible4ever
* Negative tests
  + Incorrect username and passwords to make sure not everyone can just log in
    - Ex Mark55 , wrongpass123
    - Ex aksdjfnkdsjfnsdkn, skdjfnsdkfnsdfksj

convertInchesToMeters(int inches)

positive tests

* 50 => 1.27
* 76 => 1.93

Negative test

* -3 => NegativeLengthExecption

What should be tested

* Test things that there is logic within them
* DO NOT test getters and setters that have no inner logic in them

Test terminology

* Test case is a single test.
  + It is one set of inputs
* Test Suite is a collection of test cases

Maven

* It’s a build automation and dependency management tool
  + A dependency is any piece of Java Software/ code that is not inherently in the Java Standard Library the every version of Java has
    - Some 3rd party wrote a dependency
  + We can add dependencies to our project with Maven
  + POM.xml (Product Object Model)
    - A configuration and information file for our Project
    - Contains information like
      * Dependencies
      * Author
      * Name of the application
      * Version
      * What version of Java to compile into
      * Specifics on how to build this application