Unit Testing

Objectives

- Write a program using Test Driven Development practices
- Analyze the different types of "testing"

Assignment Part 1 – Test Driven Development programming

We will be using cyber-dojo in order to have a pre-setup easy-to-use testing environment.

1. FORK from http://54.153.4.164/dashboard/show/FB71DA05B1 Follow the forking process detailed in the lab to get your own workspace.
2. You MUST follow strict Test Driven Development practices to get credit for this assignment. This will be indicated by the traffic light activity at the top of the cyber-dojo site. Make sure you are clicking "test" after each time you change the test code and after each time you change the production code so it is obvious you are following TDD.
3. You must have tests for the numbers listed below. It is recommended to write these tests one at a time in the order given. A minimum of seven cycles of "write test, tests fails, write production, tests pass" is required.
    a. 1, 2, 3
    b. 5
    c. 6, 7, 8
    d. 4
    e. 10, 13, 14, 18
    f. 9
    g. 19, 39
    h. 50, 88
    i. 40
    j. 100, 300, 149
    k. 90, 99
    l. 500
    m. 400, 499
    n. 1000, 4000
    o. 999

Keep in mind when implementing production code that you are ONLY allowed to write the minimum code required to make the tests pass. Although some solutions may be able to handle many of these cases with a relatively short amount of code, DO NOT take these routes until you have finished the TDD and finished writing all your tests. At this point you may feel free to "refactor" your code to make it more efficient/simpler/easier to read. The beauty of unit testing is that you can feel free to make as many changes to your production code as you want and as long as your tests pass you know you haven't broken any functionality.

Assignment Part 2 – Short analysis of different types of testing

1. Write a short essay analyzing the differences between the following testing strategies:

- Unit Testing
- Integration Testing
- Functional Testing
- Acceptance Testing

2. Consider a company that has many unit tests covering nearly all of the code base. Although it can take a lot of time to create many tests, what advantages does a company like this have over a company that does not practice writing test code? If you were a developer starting at one of these companies which would you feel more comfortable at? Why?

You should write at least 250 words analyzing the different testing strategies and at least 100 words discussing advantages of having a largely tested code base.

SUBMISSION

Submit to Moodle a **PDF file** of the essay with your name and cyber-dojo id and avatar at the top. Review the lab assignment if you do not remember how to get the cyber-dojo id and avatar name.