Michael Frank
CSSE 376
Lab 5

1) The first step of test driven development is to write some minimal test case. When this test fails you write the least amount of code that it takes to satisfy that test. Once the first test passes, you write another test that is slightly more complex or tests a new aspect of the function. Once that test is in place, you develop the code to pass that test. This process repeats until the function is fully functional and you have enough test cases to show that the function is reasonably correct.

2) I agree with the claim that TDD increases confidence in the code written by software developers. The tests created during developement prove that the code is functional in the areas tested. The testing process also allows for more thorough tests to be created since the developers might think of how to break their systems as they write better than when testing at the end of a cycle.

I don't agree with the claim that TDD improves overall code quality. TDD will create perfectly functional code but it won't be better code than traditionally written code. TDD allows for code to be sloppy as long as it is functional. This sloppy code can easily be fixed by refactoring, but it still might have started out more sloppy than other code.

3) TDD is superior to normal development by implementing the tests as you write. When the code has passed the last test, you know that it has been tested and is reasonably correct. TDD runs into disadvantages when faced with a problem that doesn't flow from the simple cases to the actual algorithm. The last problem could only go so far by hard coding before the actual solution had to be coded. There was no middle ground.