Elizabeth Hines
CSSE 376- Lab 5
March 22, 2012

1) TDD requires developers to write the tests for the new feature before writing the actual code. First the developer writes tests for one feature, then writes code that will pass the tests and finally refactors. This process is continued until all features are completed.
2) I do believe TDD improves the confidence of the developer because you can see that the code you are writing is doing what you thought it would the minute you write it. It does not require developers to code a whole application, then all the tests and finally test it; it is much easier to correct code when you are already in the mindset of the feature.
   I am not sold on TDD improving code quality; I think TDD makes it too easy to just code to pass a test, not necessarily code efficiently. To fix this once the test for each feature is passed the code should be refactored.
3) For such a small project, TDD was very unnecessary and made the project more time consuming. For a bigger projects I assume TDD would greatly help, providing immediate feedback for a feature after the code is written.