**Name:**

1. How long did it take to complete the complete task? Do you think this was longer or shorter than you would have completed the assignment if you did not use TDD, but did write tests after you coded?

I took 4 hours. This is probably slightly faster than without TDD, but it's hard to say.

1. Did you follow the TDD approach strictly? If not, how did you deviate from the practice?

I tried to. I found myself occasionally implementing obvious things like handling errors while working on implementing features because it was easy to do so in one or two lines (and then writing tests for those afterwards). I also sometimes wrote a more general solution the first time I did something instead of writing a naïve implementation first.

1. Did you work with a partner? If so, approximately what percent of the time did you spend as the pilot?

No.

1. Do you think your code is correct? Are you more confident that it is correct than you were with the pair programming assignment? Explain your answer to support your conclusions.

I think it is correct but I would be more confident if I had written it by thinking about how I wanted to design it and writing it that way (probably with tests written before rather than after would make me most confident in correctness, but I would be more confident if development [particularly, design] was not fully driven by the tests).

1. What problems did you encounter with TDD? What, if anything, worked well for you? Provide details.

I had a hard time forcing myself to not do obviously correct/good things (like catch an exception proplerly) and instead wait until I needed to do that to pass a test. I also like to think my design through before/during implementation and come up with a design that 'feels' elegant and nice, but TDD somewhat precludes this.

1. Would you consider making TDD a regular development practice? Why or why not? Support your conclusion.

No. It was probably marginally faster than my usual methods, but I am not convinced it is better for correctness and I do not enjoy it much. While it may produce correct code (by the tests), I believe a well-thought-out design can produce code which is more easily maintained or extended if additional functionality is needed or bugs are found.