**Name:** Tushar Narayan

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?

It took about 5 hours to complete the task. I think this was slightly less than the time it would have taken me to complete the assignment not using TDD. This is because using TDD allowed me to focus solely on what the requirements were for the functions, and once I had the tests down I could write the code for the functions. This ensured that I didn’t write any unnecessary code, and could focus on the precise specifications of the assignment.

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

I followed the TDD approach for the most part, with the sole exception of writing the checks for an input of Roman numerals for the toRoman function and the input of Arabic numbers for the toArabic function. This is because the idea that writing these checks would be a nice-to-have feature entered my mind only when writing the actual code itself, and so I didn’t have a test for those bits of code before writing the code itself.

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

I did not work with a partner.

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 my code is correct. I am definitely more confident that it is correct than I was with the pair programming assignment. This is because writing the tests allowed me to focus on what the essential features were (the checks described in Question 2 are nice-to-have’s, not essential features), and test explicitly for those; thus making sure that the code I wrote fulfilled the specifications of the assignment.

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

I found it slightly frustrating to write tests before writing the code. As a student, all my previous experience nudged me towards writing code before writing tests. However, once I got into the mode of writing tests, I found it to work well.

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

I think it would depend on the programmer. Certain programmers may not like being forced to write tests before code, and enforcing TDD would only make them resentful and less productive. However, TDD certainly has its merits and I found it a useful tool to think about specifications and features – I can thus see it being useful for projects where the specifications require some thought before code is written to ensure quality.