Tiffany Pohl
CM 784
Lab 5

1. For TDD, you write tests before you write the actual code. More specifically, you will write tests until one fails and then you will implement the program until all of your current tests past. This process repeats itself until you are satisfied all the needed tests pass. Throughout the coding process you want to refactor early and often and after each refactor, you want to make sure the code still passes the given test cases.
2. I agree with these claims. Coding the program with the goal for it to pass the given tests is a great way to code a program as it shows that you are making progress. If you wait to test until the end, there is the possibility that you will have a lot of bugs that you have to figure out at once versus finding and fixing the bugs along the way. The refactoring early and often helps the code stay clean and easy to read which is improving the overall code quality.
3. Using TDD guides your code. It gives you something to shoot for rather than just coding random things. It also allows you to see whether or not your code works. The largest disadvantage of using TDD is creating special cases then eventually combining them into one case that covers everything. Also, committing after every test was added and after each time the code was modified was a bit redundant.