The point of this project was to develop a program to compute the strongly connected components of a directed graph. We used the various graph classes that were provided and the algorithm described in class. Instead of getting the input from the command line, I ask the user for the name of the text file. So the main program asks the user until an existing file is read and then send the name of the file to the constructer. The constructer reads the file and builds the graph, prints the adjacent list and adds and prints the vertex stack. Then constructs another graph which is a copy of the first with each edge reversed., prints the reverse graph and finally prints the components.

To accomplish this project I used many methods provided by the classes in the graph file but, after using the provided DfsGraph to run it on the initial test graphs, I was to implement my own version of the DfsGraph class. So I implement the strongly connected components algorithm using my own version. In MyDfsGraph, I wrote my own constructors and firstPass, dfs, reverse, secondPass, preVisit and postVisit methods. Using these methods I was able to print out information of the inputted graphs.

I decided to make the program ask for the file name until it finds it. If the file does not exist then it will print what the user entered and ask again, so no need to rerun the program if the user entered a mistaken file name. As for the output, besides the sample output provided by the professor, I hand drew the graphs to double check I was printing the write information.

This was an interesting project. Challenging but not too hard once you draw the graphs and see what is suppose to happen. Programming the steps required was the challenging part but not impossible. I now have a better understanding of directed graphs; about their components and their adjacent list, even when reversed. Nice project to end the quarter.