# Salwa Haider CS

# 4348.001

# Summary

The project studies the coordination of multiple threads using coordination of multiple threads using semaphores. A bank is simulated by using threads and semaphores to model customer and employee behavior. This project is similar to the “barbershop” example in the textbook. Threads use semaphores for communication and mutual exclusion. In this project, I used classes called Customer, Teller, and Loan Officer to represent all the characters in bank simulation. Each class implements Runnable interface.

My approach was that I made a simple outline of character actions and followed that to come up with the pseudo code of my design. My pseudo code is more related to my thinking process and what is actually happening when the program is being run. For example, my thought process was that if I was a customer at a bank, the first thing I would want is to get in line for what I was there for. Therefore, my pseudo code for the Customer class starts with a customer entering the bank, getting a random task and amount for task assigned to them. After, the customer waits in line till the teller or the loan officer is ready to see them. I coded this using Java as I have more recent experience in Java and I feel like it is what I am better at.

When I started my project, I did not realize that I named it Customer and not Bank Simulation or Project2 so trying to make a separate class for Customer did not work out. Also, I had trouble trying to compile the program when I had each class in a different file, so I combined all of it into one file – meaning all three classes and the main function is in a single program. Moreover, it took me a lot of internet search and taking help from the book to figure out how to get all the threads to work with each other and how to signal and wait semaphores on each other.

I learnt how to create and use semaphores, the use of signal and wait and creating threads and getting them to work with one another. It was a good project for me to even learn how they are working practically.