1. This project is a scheduling program focused on allowing the user to enter their classes and schedule requirements, and then utilizes these to create a randomized schedule.The program uses user-input to create a **plethora** of different schedules. It is tasked to create every iteration that is possible for within the constraints of the class and academic requirements given to the program by the user. After it creates the schedules, it prints them to an easy to read text file so the user can best see the different schedules created. The user then has the job of selecting the schedule they like best.

2. We primarily used void/string/int functions for our program, and conditionals (while loops and if/else loops) along with file I/O. Another aspect we used was the switch command to help better process user input. We selected these structures because without the functions, if we were to enter a class, reading each line in the file to check the requirements without set conditions would make processing the file difficult at best. Also, since these are simple structures, it is easier to experiment and test the code as well. We chose file I/O because we wanted to make the program flexible enough to fit any major available. This allowed us to create an adaptable output file unique to the user, and a function that would use that file to select a schedule at random. This is done by by utilizing the specific characters in a string found in the line also containing the course name.

3.A step-by-step instruction of how to use the code. This should include a test case for running your code.

Warning: When the program begins, you MUST make sure to follow the instructions by the letter. Mistakenly entering a key that is meant to be in CAPs may crash the program. Mistakenly pressing ESC in parts of it, may crash the program. Furthermore, where the program asks for set number of courses, that is the limit -- every course counts, including the co/prerequisites not listed as requirements. {this is the counter part that’s irritating me}

1. Click on the .exe file to open the program. You will be greeted by a prompt …. (word what you see as you like)
2. To quit the program, use ESC. To use it, hit ENTER.
3. If you hit enter, the program explains what it is doing and provides the user with instructions.
4. Normally these will either ask you to type something into the console, or to read the instructions on the screen. Do so.
5. Repeat step 4 until the class schedule is built.
6. Once it is built and the program completes the rest, open the same folder that contains the .exe file for the program, and open reqs.txt.

4. IDE(s) and OS(s) used:

Development: Codeblocks on Windows 10, Visual Studio Enterprise on Windows 7.

Testing: Codeblocks on Windows 10, Visual Studio Enterprise on Windows 7.

5. If you worked with a partner, your documentation must also include a listing of who did what.

**Diana Carcamo** wrote and adapted the functions to work with the current iteration of the code, and focused on editing the project.

**James Kuczynski** created the initial implementation of pseudocode, helped with the writing and drafting of additional pieces to Diana’s code.