Schedule Creator C++ Project Summary

The primary purpose of my program is to eliminate much of the time I waste *attempting* to organize my time in my head by having a simple handy tool that will do it for me. The idea is that every time I am assigned a homework assignment, reading, or project in class, I can simply add a line to a .csv file and the program will figure out when I should complete it.

A line of the file might look like this: `CSCI2824 HW11,04/19/19,5,12.00,180,1`

Reading from this file, the program uses a Heap to prioritize the tasks I have to complete. It also reads recurring “events” such as classes, or one time “events” such as appointments from .csv files, so that it knows what times to work around.

The primary functions to go about this are found in the Schedule.cpp class, and utilize other functions from the SC_TaskHeap.cpp file in order to sort tasks, SC_Helpers.cpp file in order to properly deal with messy data items such as “Dates”, and the SC_Structs.cpp file, which contains Date, Event, and Task data structure information.

There are 7 menu options.

The first 3 are mainly to showcase the functionality of the Heap implementation. When I alter the program for personal use and tweak the prioritization algorithms, I may not even include them. These 3 allows you to see the most prioritized task, add a new task to the heap, and complete the first task in the heap, but none of the information will be saved when the program is quit.

The 4th menu item prints out all tasks in the heap, including tasks that will not be saved when the program is quit. It also showcases the Heap implementation, but also

practically allows you to review your prioritized list of tasks in a formatted fashion (unlike the unsightly .csv file)

The 5th menu item prints the whole schedule in the console but does not save it. This allows the user to decide if he or she is happy with the proposed schedule. If not, the user can “add a task” with a certain due date and name it “free time”, thereby adjusting the schedule output.

The 6th option writes the most up to date schedule into a text file.

The 7th option asks the user if he or she would like to save the schedule (option 6), then quits the program.

Note: I skipped the “title screen” and date and time retrieval steps for the example outputs to make all of our lives easier. The file titled “ExampleOutputFull.rtf” contains the output for the beginning of the program. If you run it, these 2 steps will not be omitted. Hope that’s okay.