Nicole Bowler

CS 202 - 1001

Due Wednesday, February 7th, 2018

Christos Papachristos

Project II

The intent of this project is to let us use and get familiar with structs (and specifically arrays of structs) as well as working with pointers and passing arguments into functions by reference while still designing and implementing a solution to a problem through the C++ language. Other previous concepts such as arrays, file I/O, console I/O, and character arrays are also included. The problem given is to write a program that asks the user to input a file of five cars with multiple attributes of multiple data types and to store those data types into a struct defined as having an *int* for the 'year', two *c-strings* for the 'make and model', a *float* that represents the 'price per day", and a *bool* that dictates whether or not the car is available. This program also includes a menu system with 7 options: 1) enter a new input file, 2) print out the struct to the console, 3) print out the struct to a user defined output file, 4) sort cars by price, 5) show the user the total price of the available cars based on the number of days entered at the beginning of the function, 6) have a car be rented by checking if it is available and changing the *bool* value accordingly, 7) exit the program.

My solution was to have each option be a case of a switch statement that ran in a loop until option 7(exit) was entered. The menuNum int stored the option and used it as the expression for the switch statement. Each case for the switch consisted only a function and break statement to prevent the program falling through each case. The functions performed each menu

option independently and would return an *int.* The int would come from a sub-function called 'printMenu.' This supplementary function is found in every menu function and it would print out the menu to the console for the user, ask the user to enter an option and then return that option to the larger function which would then immediately take that value and return it to menuNum. This meant that after every menu option was executed, the menu would be printed and a number for the switch statement would be ready by the time the switch statement got executed in the main loop.

One major bug that I could not resolve due to time constraints was found in sortCar(). Luckily since this function is modular and completely independent, it only affects the struct array if called upon. The function would sort the array of structs but would do so that garbage would be left in for all of the data types in the structs. This could possibly be from undeclared arrays being assigned to existing arrays but after several hours of debugging and with little time left to submit, I could not find where the bug was located. If I had more time on this project I would resolve this issue. I would also test the program more as it is possible for a plethora of inputs to be made in various combinations and each one needs to be as consistent as the others. A minor addition I wish to also make to the code is in the exit function. Before the program returns 0 to the compiler, it closes all open files. To make sure that happens every time I would like to add a message printed to console saying something along the lines of "Exiting program…"