-----------------------------------------------------------------------------------------------------------------------

Name: Hashmeet Singh Saini

Seneca ID : hsaini28

Student No.: 153070214

Date : Septemeber 12th, 2022

I have done all the coding by myself and only copied the code

that my professor provided to complete my project milestones.

------------------------------------------------------------------------------------------------------------------------

This week’s workshop was a light revision of some topics from OOP244 and also introduced some new topics like Internal and External linkage, use of the main function’s overload that receives command line arguments, use of static and constexpr keywords and more. The keyword constexpr has been used in line 12 of the w1\_p2 file because it shifts the expression calculation from run-time to compile-time making a program a little faster and more efficient.

1. Internal linkage is when two names are used in different scopes of the same module. The keyword “static” is used to achieve this linkage. It was used in line 102 to create a variable called counter. On the other hand, External Linkage is when 2 names belong to two different scopes and different modules. Keyword ‘extern’ is used for achieving this. It was used to create 2 variables called g\_taxrate and g\_discount in carads.h file to access these variables in carads.cpp and w1\_p2.cpp files.
2. Static variables are the variables that are retained in the memory throughout the lifetime of a program. In my code, it was used in file carads.cpp line number 102 to create a variable called counter in the display function. This helped me to take care of the serial number while displaying records as every time the function was called, the variable never went out of scope and the serial number could be easily incremented.
3. When the Cars class was upgraded, I modified the m\_brand variable to be able to hold a flexible character array of brand name through dynamic allocation. Since dynamic allocation comes with its own requirements, I also had to create a destructor for deleting memory on heap and copy constructor and copy assignment operators in order to prevent any kind of possible memory leaks. I also had to make changes to the read function and the >> operator overload to be able to correctly allocate dynamic memory.