Name: Jingmin Zhou (Jessica)

Student ID: 119766194

Email: [jzhou175@myseneca.ca](mailto:jzhou175@myseneca.ca)

In the Workshop 3 (Classes & Privacy), I have learned how to define a class type and private its data members and public its member functions as needed, how to instantiate an object of class type, how to access private data and change the data values within an object of class type through public member functions, like get() and set() functions. Additionally, I learned to format the output using standard library facilities, like cout.setf(iso::fixed).

Firstly, when we set Train or Cargo data members, we reject the given value set default value for corresponding data member if the given values are invalid, which we call as empty state. Using empty state, we can validate information incoming from client code before storing it in an object. The possible empty state values could be nullptr for character pointer, empty string for string pointed to by the character pointer, and 0 for integer type ID number.

Secondly, data members defined are exposed to client code using the struct keyword, and any client code can change the value of a data member, while data members defined are hiding to client code using the class keyword.

Finally, class privacy means the private members defined in class are not accessible outside the class and can be accessed only through member functions of the class. The public members, like member functions, form an interface to the class and are accessible outside the class. Data privacy is important to encapsulation in object-oriented programming. By class privacy through hiding all data members from client code, we can not only prohibit the client to access the data directly for data safety, but also control over which data to accept, which data to reject and which data to expose and make sure the safe data status.