// OOP345 Project

// File: reflect.txt

// Name: Jingmin (Jessica) Zhou

// Email: jzhou175@myseneca.ca

// ID: 119766194

// Date: 07/30/2020

In the project, a simulated Assembly Line was developed based on C++ Object Oriented Programming. The simulated assembly line fills customer orders from inventory. The line consists of a set of stations (Station/Workstation Module), each of which holds an inventory of items for filling customer orders (CustomerOrder Module) and a queue of orders to be filled and fills the next order in the queue if that order requests its item and that item is still in stock. A line manager (LineManager Module) moves the customer orders from station to station until all orders have been processed.

Firstly, a supporting class Utilities was used to populate string data from the input files into specific objects (e.g. Station and CustomerOrder) from text files (e.g. Station1.txt and CusotmerOrders.txt). I learned how to read the data from files through file stream mode and handle the reading exceptions.

Secondly, all the outputs (e.g. Station::display(), CustomerOrder::display() and LineManager::display()) in the project are formatted through setting the width of fields and alignment (<iomanip> library). For example, static variable m\_widthField was used to print to screen the *item name*, *serial number* and *quantity* for any object of type Station.

Thirdly, in LineManager module, displayStation() and displayStationSorted() are implemented based on two different data structures, vector and linked list respectively. Function displayStation() displayed the Workstation one by one, while displayStationSorted() displayed the linked Workstation on the assembly line, which consists of this kind of Workstation struct, Workstation { Workstation v, Workstation\* next\_w}.