**Project Overview**

The purpose of this term project is to put your C++ Object Oriented skills to practice by developing a simulated Assembly Line. The project is broken down into 3 milestones to help guide you through the process of implemented, debugging and execution a more complex application than a simple lab/workshop.

**Milestone #1 – Utilities and Item class**

This milestone will implement and test the Utilities and Item classes. The Utilities class is an object used to support the parsing of input files to setup and configure the assembly line simulation. The Item class encapsulates the information about a given Item on the assembly line that can be filled within a customer order. The definitions of the classes are defined below.

**Starting Point**

To start, download the Visual Studio solution space which provides the starting point and main program for all milestones in the project. Once you unzip this solution, you will noticed all the headers, inputs files and AssemblyLine.cpp files are present. Your job is to create the class implementation files (\*.cpp) as per the specifications provided.

**Utilities Class**

Parsing string data from input files is performed the same for all objects within the system. The **Utilities** object provides the basic functionality required for all objects in the assembly line system. Each **Utilities** object holds the following information:

* **Field\_width** specifying the length of the token extracted – used for display purpose later
* **Delimiter** used to extract tokens from a given std::string

*NOTE: All* **Utilities** *objects share the same* **delimiter***.*

**Constructor**

Upon instantiation, a **Utilities** object initializes the value of its **field\_width** instance variable to 1.

**Member Functions**

The member functions of the **Utilities** class include:

* **void setFieldWidth(size\_t fw)** - sets the field width of the current object to **fw**
* **size\_t getFieldWidth() const** - returns the field width of the current object
* **const std::string extractToken(const std::string& str, size\_t& next\_pos, bool& more)** -- This function uses the **delimiter** to extract the next token from **str** starting at **next\_pos**. If successful, it returns the extracted token found and sets **more** to true; false otherwise.  This function reports an exception if there are two delimiters with no token between them. This function updates the current object's **field\_width** data member if its current value is less than the size of the token extracted.
  + NOTE: **str** represents a single line read from the task.txt input file
* **static void setDelimiter(const char)** - sets the delimiter for this class to the character received
* **const char getDelimiter() const** – returns the delimiter character of the current object.

**Item Class**

An **Item** object manages a single **Item** on the assembly line. Each **Item** holds the following information:

* **name** – the name of the **Item**
* **serialNumber** – the next serial number to be assigned on the assembly line
* **Quantity –** the number of current items left in stock
* **Description –** description of the item
* **field\_width** – the maximum character width used for display purposes later

*NOTE: All* ***Item*** *objects share the same* ***field\_width***

**Constructor**

Upon instantiation, an **Item** object receives a reference to an unmodifiable **std::string**.  This string contains a single record (one line) that has been retrieved from the *inventory.txt* file specified by the user.

The constructor then uses a **Utilities** object to extract each token from the record and populates the **Item** object accordingly.

*NOTE: You can assume that a record from the task file is formatted as follows:*

*<name of item>|<starting serial number>|<quantity in stock>|<description>*

Once the constructor has extracted all of the tokens from the record, it validates the **Item field\_width** with the **Utilities field\_width**. If **Utilities field\_width** is greater, reset the **Item field\_width** to the **Utilities field\_width**.

*NOTE: The display member function uses this field width to align the output across all of the records retrieved from the task file.*

**Member Functions**

The normal member and helper functions of the **Item** class include:

* **const std::string& getName() const** - returns the name of the current Item object
* **const unsigned int getSerialNumber()** – returns the next serial number to be used on the assembly line
* **const unsigned int getQuantity()** – returns the remaining quantity of the current Item object
* **void updateQuantity()** – validates if the quantity is greater than zero and decreases the quantity by one.
* **void display(std::ostream& os, bool full) const** - writes the **name** and **serialNumber** of the Item object to the **os.** If full status is requested, it will include the **quantity** and **description** of the **Item** object

**Milestone #1 – Sample Output**

The output for this milestone should look as follows:

CPU [123456] Quantity 5 Description: Central Processing Unit

Memory [654321] Quantity 10 Description: Basic Flash Memory

GPU [456789] Quantity 2 Description: General Porcessing Unit

SSD [987654] Quantity 5 Description: Solid State Drive

Power Supply [147852] Quantity 20 Description: Basic AC Power Supply

Manual Validation

getName(): CPU

getSerialNumber(): 123456

getSerialNumber(): 123457

getQuality(): 5

getQuality(): 4

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*