**CSC 1100 – Problem Solving and Programming**

**Project 2 – Omar Faruk**

**50 points – Due December 17, 11am**

**Late deadline is December 19, 11:59pm, but 20% off**

**a)** Save this document with your name and the project number somewhere in the file name.

**b)** Paste your code and screenshots into the document.

**c)** Submit this document and your .cpp file(s) to the Canvas item where you downloaded this document. Do not submit a zip file but individually attach your files.

Remember that great C++ console application you wrote for *Snow Shepherds* (Lab 22). It searches a product inventory. Now they need you to add the following four options:

4-Change wholesale price

5-Change retail price

6-Order product

7-Sell product

Present the following menu to the user:

Snow Shepherds Menu

1-Find product by ID

2-Find product by Name

3-List products

4-Change wholesale price

5-Change retail price

6-Order product

7-Sell product

9-Exit

Enter an option: "

Add the following options:

**Option 4 – Change wholesale price**

This option prompts the user for an ID, and searches the products array for the ID using function **findProductByID**. If the ID is not found, it prints an error message. If the ID is found, it uses a validation loop to prompt for and get a new wholesale price for the product that is greater than zero. It stores the new wholesale price and prints a message.

**Option 5 – Change retail price**

This option prompts the user for an ID, and searches the products array for the ID using function **findProductByID**. If the ID is not found, it prints an error message. If the ID is found, it uses a validation loop to prompt for and get a new retail price for the product that is greater than the wholesale price. It stores the new retail price and prints a message.

**Option 6 – Order product**

This option prompts the user for an ID, and searches the products array for the ID using function **findProductByID**. If the ID is not found, it prints an error message. If the ID is found, it uses a validation loop to prompt for and get a quantity to order for the product that is greater than zero. It adds the quantity to the existing quantity for the product and prints a message.

**Option 7 – Sell product**

This option prompts the user for an ID, and searches the products array for the ID using function **findProductByID**. If the ID is not found, it prints an error message. If the ID is found, it uses a validation loop to prompt for and get a quantity to sell for the product that is less than the existing quantity. It subtracts the quantity from the existing quantity for the product and prints a message.

Insure that your code is commented! Provide a complete header comment and body comments. Again, define constants for the array size, input file name, and column widths. Continue to process menu options until the user enters option 9. Note that the four new options are fairly similar. Code and test one of them, and then use that code as a basis for the other options. The output should look like this:

Welcome to Snow Shepherds, v2

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

5 line(s) read from file 'SnowShepherdsInventory.txt'.

Snow Shepherds Menu

1-Find product by ID

2-Find product by Name

3-List products

4-Change wholesale price

5-Change retail price

6-Order product

7-Sell product

9-Exit

Enter an option: 3

Product Inventory

Code Product Wholesale ($) Retail ($) Inventory

24 Boots 140.00 260.75 23

17 Skis 300.50 550.55 8

39 Hat 15.25 30.20 42

71 Gloves 18.10 28.00 51

65 Snowboard 400.40 620.50 18

Snow Shepherds Menu

1-Find product by ID

2-Find product by Name

3-List products

4-Change wholesale price

5-Change retail price

6-Order product

7-Sell product

9-Exit

Enter an option: 4

Enter product ID to change wholesale price for: 25

Error: product ID '25' not in inventory.

Snow Shepherds Menu

1-Find product by ID

2-Find product by Name

3-List products

4-Change wholesale price

5-Change retail price

6-Order product

7-Sell product

9-Exit

Enter an option: 4

Enter product ID to change wholesale price for: 17

Enter new wholesale price (currently $300.50): -25

Error: wholesale price must be greater than zero.

Enter new wholesale price (currently $300.50): 225

Wholesale price changed to $225.00.

Snow Shepherds Menu

1-Find product by ID

2-Find product by Name

3-List products

4-Change wholesale price

5-Change retail price

6-Order product

7-Sell product

9-Exit

Enter an option: 5

Enter product ID to change retail price for: 71

Enter new retail price (currently $28.00): 12

Error: retail price must be greater than wholesale price (currently $18.10):

Enter new retail price (currently $28.00): 24

Retail price changed to $24.00.

Snow Shepherds Menu

1-Find product by ID

2-Find product by Name

3-List products

4-Change wholesale price

5-Change retail price

6-Order product

7-Sell product

9-Exit

Enter an option: 3

Product Inventory

Code Product Wholesale ($) Retail ($) Inventory

24 Boots 140.00 260.75 23

17 Skis 225.00 550.55 8

39 Hat 15.25 30.20 42

71 Gloves 18.10 24.00 51

65 Snowboard 400.40 620.50 18

Snow Shepherds Menu

1-Find product by ID

2-Find product by Name

3-List products

4-Change wholesale price

5-Change retail price

6-Order product

7-Sell product

9-Exit

Enter an option: 6

Enter product ID to order: 39

Enter quantity to order (currently 42): 5

Quantity in stock is now 47.

Snow Shepherds Menu

1-Find product by ID

2-Find product by Name

3-List products

4-Change wholesale price

5-Change retail price

6-Order product

7-Sell product

9-Exit

Enter an option: 7

Enter product ID to sell: 65

Enter quantity to sell (currently 18): 22

Error: quantity must be less than or equal to 18.

Enter quantity to sell (currently 18): 6

Quantity in stock is now 12.

Snow Shepherds Menu

1-Find product by ID

2-Find product by Name

3-List products

4-Change wholesale price

5-Change retail price

6-Order product

7-Sell product

9-Exit

Enter an option: 3

Product Inventory

Code Product Wholesale ($) Retail ($) Inventory

24 Boots 140.00 260.75 23

17 Skis 225.00 550.55 8

39 Hat 15.25 30.20 47

71 Gloves 18.10 24.00 51

65 Snowboard 400.40 620.50 12

Snow Shepherds Menu

1-Find product by ID

2-Find product by Name

3-List products

4-Change wholesale price

5-Change retail price

6-Order product

7-Sell product

9-Exit

Enter an option: 9

End of Snow Shepherds, v2

Do not use this sample input for the final run pasted below.

*[your program code here]\**

**//==========================================================**

**//**

**// Title: Snow Shepherds Inventory Application v2**

**// Course: CSC 1101**

**// Work: Project 2**

**// Author: Omar Faruk**

**// Date: 12/13/2020**

**// Description:**

**// V1:**

**// Creating a console application to utilize Snow Shepherds**

**// inventory by allowing user to find product using search ID and name**

**// and also listing the full inventory. Tools used were functions, structs,**

**// loops, branching, and utilizing an input file to read data from.**

**//**

**// V2:**

**// Added menu options 4 to change wholesale price, 5 to change product price,**

**// 6 to order product, and 7 to sell products by adding additional else if statements and options to menuOption’s function**

**//==========================================================**

**#include <cstdlib> // For several general-purpose functions**

**#include <fstream> // For file handling**

**#include <iomanip> // For formatted output**

**#include <iostream> // For cin, cout, and system**

**#include <string> // For string data type**

**using namespace std; // So "std::cout" may be abbreviated to "cout"**

**// Globals**

**const int ARRAY\_SIZE = 5;**

**const int COLMFT1 = 20, COLMFT2 = 7;**

**// Declare product struct**

**struct product**

**{**

**int product\_ID;**

**string product\_name;**

**double wholesale\_price;**

**double retail\_price;**

**int product\_quanitity;**

**};**

**// Menu Option**

**int menuOption()**

**{**

**// Declare variables**

**int option;**

**// Show menu and get option**

**cout << endl**

**<< "Snow Shepherds Menu" << endl**

**<< "1 - Find product by ID" << endl**

**<< "2 - Find product by Name" << endl**

**<< "3 - List Products" << endl**

**<< "4 - Change wholesale price" << endl**

**<< "5 - Change retail price" << endl**

**<< "6 - Order product" << endl**

**<< "7 - Change wholesale price" << endl**

**<< "9 - Exit" << endl << endl**

**<< "Enter an option: ";**

**cin >> option;**

**// Return dimension**

**return option;**

**}**

**// Find product by ID, Name functions**

**int findProductByID(product products[ARRAY\_SIZE], int key)**

**{**

**// Declare variables**

**int index = 0;**

**// Linear search product ID's & Return -1 if found**

**while (index < ARRAY\_SIZE && products[index].product\_ID != key)**

**{**

**index = index + 1;**

**}**

**if (products[index].product\_ID == key)**

**return index;**

**else**

**return -1;**

**}**

**int findProductByName(product products[ARRAY\_SIZE], string key)**

**{**

**// Declare variables**

**int index = 0;**

**// Linear search product ID's & Return -1 if found**

**while (index < ARRAY\_SIZE && products[index].product\_name != key)**

**{**

**index = index + 1;**

**}**

**if (products[index].product\_name == key)**

**return index;**

**else**

**return -1;**

**}**

**//Print the product function**

**void printProduct(product products[ARRAY\_SIZE], int index)**

**{**

**cout << "Product " << products[index].product\_ID << endl**

**<< setw(COLMFT1) << left << "Product:"**

**<< setw(COLMFT2) << right << products[index].product\_name << endl**

**<< setw(COLMFT1) << left << "Wholesale ($):"**

**<< setw(COLMFT2) << right << products[index].wholesale\_price << endl**

**<< setw(COLMFT1) << left << "Retail ($):"**

**<< setw(COLMFT2) << right << products[index].retail\_price << endl**

**<< setw(COLMFT1) << left << "Inventory:"**

**<< setw(COLMFT2) << right << products[index].product\_quanitity << endl**

**<< endl;**

**}**

**int main()**

**{**

**// Declare Constants**

**const string FILE\_NAME\_INPUT = "SnowShepherdsInventory.txt";**

**// Declare variables**

**ifstream SnowShepherdsInventory\_FileInput;**

**string line, product\_name;**

**int option, index, product\_id;**

**double NewPrice\_wholesale, NewPrice\_retail;**

**int line\_count = 0, order\_quantity = 0, sell\_quantity = 0;**

**// Declare 5 element array from struct**

**product products[ARRAY\_SIZE];**

**// Declare**

**cout << fixed << setprecision(2) << endl;**

**// Show application header**

**cout << "Welcome to Snow Shepherds! v2" << endl;**

**cout << "-----------------------------" << endl;**

**// Attempt to open input file**

**SnowShepherdsInventory\_FileInput.open(FILE\_NAME\_INPUT);**

**if (!SnowShepherdsInventory\_FileInput.is\_open())**

**cout << "Error: unable to open file '" << FILE\_NAME\_INPUT << "'." << endl << endl;**

**else**

**{**

**// Reading past header comment**

**getline(SnowShepherdsInventory\_FileInput, line);**

**while (SnowShepherdsInventory\_FileInput.good())**

**{**

**// Reading tokens from input file**

**SnowShepherdsInventory\_FileInput >> products[line\_count].product\_ID;**

**SnowShepherdsInventory\_FileInput >> products[line\_count].product\_name;**

**SnowShepherdsInventory\_FileInput >> products[line\_count].wholesale\_price;**

**SnowShepherdsInventory\_FileInput >> products[line\_count].retail\_price;**

**SnowShepherdsInventory\_FileInput >> products[line\_count].product\_quanitity;**

**// Line count in input file**

**line\_count = line\_count + 1;**

**}**

**// Close input file and output line read**

**SnowShepherdsInventory\_FileInput.close();**

**cout << endl << line\_count << " line(s) read from file '"**

**<< FILE\_NAME\_INPUT << "'." << endl;**

**option = menuOption();**

**while (option != 9)**

**{**

**// Finding product by ID search**

**if (option == 1)**

**{**

**cout << "Enter product ID to search for: ";**

**cin >> product\_id;**

**index = findProductByID(products, product\_id);**

**if (index == -1)**

**cout << "Error: Product ID '" << product\_id << "' is not in inventory" << endl;**

**else**

**printProduct(products, index);**

**}**

**// Finding product by name search**

**else if (option == 2)**

**{**

**cout << "\nEnter product name to search for: ";**

**cin >> product\_name;**

**index = findProductByName(products, product\_name);**

**if (index == -1)**

**cout << "Error: Product name '" << product\_name << "' is not in inventory" << endl;**

**else**

**printProduct(products, index);**

**}**

**// Displaying all products**

**else if (option == 3)**

**{**

**cout << setw(COLMFT1) << left << "\nCode"**

**<< setw(COLMFT1) << left << "Product"**

**<< setw(COLMFT1) << left << "Wholesale ($)"**

**<< setw(COLMFT1) << left << "Retail ($)"**

**<< setw(COLMFT1) << left << "Inventory" << endl;**

**for (int i = 0; i < ARRAY\_SIZE; i++)**

**{**

**cout << setw(COLMFT1) << left << products[i].product\_ID**

**<< setw(COLMFT1) << left << products[i].product\_name**

**<< setw(COLMFT1) << left << products[i].wholesale\_price**

**<< setw(COLMFT1) << left << products[i].retail\_price**

**<< setw(COLMFT1) << left << products[i].product\_quanitity**

**<< endl;**

**}**

**}**

**// Setting new wholesael eprice**

**else if (option == 4)**

**{**

**cout << "Enter product ID to change wholesale price for: ";**

**cin >> product\_id;**

**index = findProductByID(products, product\_id);**

**if (index == -1)**

**cout << "Error: Product ID '" << product\_id << "' is not in inventory" << endl;**

**else**

**{**

**// Validation loop for new user input**

**cout << "Enter new wholesale price (currently $" << products[index].wholesale\_price << "): ";**

**cin >> NewPrice\_wholesale;**

**while (NewPrice\_wholesale < 0)**

**{**

**cout << "Error: wholesale price must be greater than zero" << endl**

**<< "\nEnter new wholesale price (currently $" << products[index].wholesale\_price << "): ";**

**cin >> NewPrice\_wholesale;**

**}**

**// Setting the price with confirmation output**

**products[index].wholesale\_price = NewPrice\_wholesale;**

**cout << "Wholesale price changed to $" << products[index].wholesale\_price << "." << endl;**

**}**

**}**

**// Changing the retail price**

**else if (option == 5)**

**{**

**cout << "Enter product ID to change retail price for: ";**

**cin >> product\_id;**

**index = findProductByID(products, product\_id);**

**if (index == -1)**

**cout << "Error: Product ID '" << product\_id << "' is not in inventory" << endl;**

**else**

**{**

**// Validation loop for new user input**

**cout << "Enter new retail price (currently $" << products[index].retail\_price << "): ";**

**cin >> NewPrice\_retail;**

**while (NewPrice\_retail < products[index].wholesale\_price)**

**{**

**cout << "Error: retail price must be greater than retail price $" << products[index].retail\_price << "." << endl**

**<< "\nEnter new retail price (currently $" << products[index].retail\_price << "): ";**

**cin >> NewPrice\_retail;**

**}**

**// Setting the price with confirmation output**

**products[index].retail\_price = NewPrice\_retail;**

**cout << "Retail price changed to $" << products[index].retail\_price << "." << endl;**

**}**

**}**

**// Ording a product**

**else if (option == 6)**

**{**

**cout << "Enter product ID to order: ";**

**cin >> product\_id;**

**index = findProductByID(products, product\_id);**

**if (index == -1)**

**cout << "Error: Product ID '" << product\_id << "' is not in inventory" << endl;**

**else**

**{**

**// Validation loop for new user input**

**cout << "Enter quanitity to order (currently " << products[index].product\_quanitity << "): ";**

**cin >> order\_quantity;**

**while (order\_quantity < 0)**

**{**

**cout << "Error: Quanitity ordered must be greater than zero" << endl**

**<< "\nEnter quanitity to order (currently " << products[index].product\_quanitity << "): ";**

**cin >> order\_quantity;**

**}**

**// Setting the quantity with confirmation output**

**products[index].product\_quanitity = products[index].product\_quanitity + order\_quantity;**

**cout << "Quanitity in stock is now " << products[index].product\_quanitity << "." << endl;**

**}**

**}**

**// Selling a product**

**else if (option == 7)**

**{**

**cout << "Enter product ID to sell: ";**

**cin >> product\_id;**

**index = findProductByID(products, product\_id);**

**if (index == -1)**

**cout << "Error: Product ID '" << product\_id << "' is not in inventory" << endl;**

**else**

**{**

**// Validation loop for new user input**

**cout << "Enter quanitity to sell (currently " << products[index].product\_quanitity << "): ";**

**cin >> sell\_quantity;**

**while (sell\_quantity > products[index].product\_quanitity)**

**{**

**cout << "Error: Quanitity sold must be less than or equal to wholesale price "<< products[index].product\_quanitity << "." << endl**

**<< "\nEnter quanitity to sell (currently " << products[index].product\_quanitity << "): ";**

**cin >> sell\_quantity;**

**}**

**// Setting the quantity with confirmation output**

**products[index].product\_quanitity = products[index].product\_quanitity - sell\_quantity;**

**cout << "Quanitity in stock is now " << products[index].product\_quanitity << "." << endl;**

**}**

**}**

**// Error menu option selected by user**

**else**

**{**

**cout << "Error: Invalid menu option, of '"<< option << "' try again!" << endl;**

**}**

**// Menu Option**

**option = menuOption();**

**}**

**}**

**// Show application close**

**cout << "\nEnd of Snow Shepherds v2" << endl;**

**}**

*[your program output here]\*\**





