**CSC 1101 – Problem Solving and Programming Laboratory**

**Lab 22 – Omar Faruk**

**25 points – Due December 7, 11pm**

**a)** Save this document with your name and the homework 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.

You've been hired by *Snow Shepherds* to write a C++ console application that searches a product inventory. Declare struct **product** with the following members:

ID – integer

product – string

wholesale – double

retail – double

quantity – integer

Declare an array of data type **product** with five elements. Attempt to open input file **SnowShepherdsInventory.txt**. The file looks like this:

ID Product Wholesale Retail Quantity

24 Boots 140.00 260.75 23

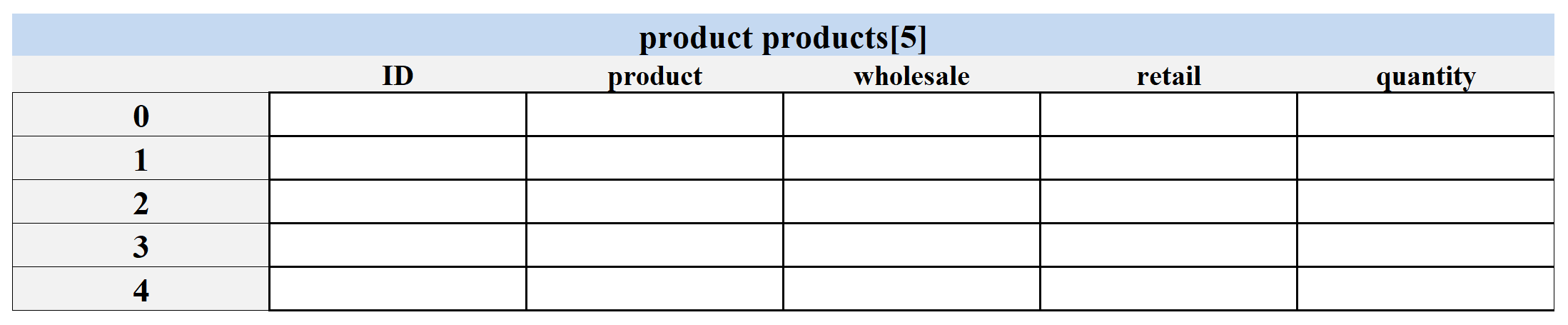
17 Skis 300.50 550.55 8

39 Hat 15.25 30.20 42

71 Gloves 18.10 28.00 51

65 Snowboard 400.40 620.50 18

It contains a header row and five detail rows, one for each product in their small inventory. If the file doesn't open, print an error message. If the file does open, reach each detail line value into the appropriate spot in the products array. The array looks like this in memory:



This spot is **products[3].wholesale**. It should contain **18.10** when file reading is completed. Present the following menu to the user:

Snow Shepherds Menu

1-Find product by ID

2-Find product by Name

3-List products

9-Exit

Enter an option: "

Create the following options:

**Option 1**

This option prompts the user for an ID, searches the products array for the ID using function **findProductByID** (described below), and prints an error message or prints product information using function **printProduct** (described below).

**Option 2**

This option prompts the user for a product name, searches the products array for the name using function **findProductByName** (described below), and prints an error message or prints product information using function **printProduct** (described below).

**Option 3**

This option prints a Product Inventory title and column headers, and loops through the products array and prints product information in columns using formatted output.

Create the following functions:

**int menuOption()**

This value function presents the menu to the user and returns a menu option.

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

This value function uses the key to linear search the product IDs. If the ID is found, it returns the index where it exists in the products array. If the ID is not found, it returns -1.

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

This value function uses the key to linear search the product names. If the name is found, it returns the index where it exists in the products array. If the name is not found, it returns -1.

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

This void function prints the product information that exists in the products array at the given index. Print the product ID as a title and then use formatted output manipulators (setw, left/right) to print the following rows:

● Product

● Wholesale price ($)

● Retail price ($)

● Inventory

And two columns:

● A left-justified label.

● A right-justified value.

Define constants for the array size, input file name, and column widths. Continue to process menu options until the user enters option 9. The output should look like this:

Welcome to Snow Shepherds

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

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

Snow Shepherds Menu

1-Find product by ID

2-Find product by Name

3-List products

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

9-Exit

Enter an option: 1

Enter product ID to search for: 33

Error: product ID '33' not in inventory.

Snow Shepherds Menu

1-Find product by ID

2-Find product by Name

3-List products

9-Exit

Enter an option: 1

Enter product ID to search for: 39

Product 39

Product: Hat

Wholesale price ($): 15.25

Retail price ($): 30.20

Quantity in stock: 42

Snow Shepherds Menu

1-Find product by ID

2-Find product by Name

3-List products

9-Exit

Enter an option: 2

Enter product name to search for: Poles

Error: product name 'Poles' not in inventory.

Snow Shepherds Menu

1-Find product by ID

2-Find product by Name

3-List products

9-Exit

Enter an option: 2

Enter product name to search for: Skis

Product 17

Product: Skis

Wholesale price ($): 300.50

Retail price ($): 550.55

Quantity in stock: 8

Snow Shepherds Menu

1-Find product by ID

2-Find product by Name

3-List products

9-Exit

Enter an option: 9

End of Snow Shepherds

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

*[your program code here]\**

//==========================================================

//

// Title: Inventory Application

// Course: CSC 1101

// Lab Number: 22

// Author: Omar Faruk

// Date: 12/07/2020

// Description:

// 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.

//

//==========================================================

#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;

cout << "Snow Shepherds Menu" << endl;

cout << "1 - Find product by ID" << endl;

cout << "2 - Find product by Name" << endl;

cout << "3 - List Products" << endl;

cout << "9 - Exit" << endl << endl;

cout << "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;

int line\_count = 0;

// Declare 5 element array from struct

product products[ARRAY\_SIZE];

// Show application header

cout << "Welcome to Snow Shepherds!" << 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)

{

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);

}

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);

}

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;

}

}

else

{

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

}

// Menu Option

option = menuOption();

}

}

// Show application close

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

}

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

