**Lab 4**

# 50 points

**(Note: This is considered a challenging lab)**

Purpose: To get experience using structures and binary files.

For this lab, please write a program that uses a structure to store the following inventory data in a file. The structure definition is provided in the starter file:

1. Item Description (Must be a char array)
2. Quantity on Hand (int)
3. Wholesale Cost (double)

You must use the starter program listed on page 2. This file is provided. You should not add anything to main. Just write the code for:

1. addRecord (**10 points**)
2. viewRecord (**10 points**)
3. changeRecord (**10 points**)

Input validation for Item Quantity and Cost: (**15 points**)

1. Must be numeric (5 **points each**)
2. Must not be negative (5 **points each**)
3. Must be greater than zero (5 **points each**) There is no validation for the Item Description.

# File specifications: (3 points)

You must submit the test file (with reasonable inventory names) that you created while testing your program. The test file should have at least 5 entries and have an extension of **“.dat”**

**\*\*\* The file format MUST be binary\*\*\*.**

**On Exit: (2 Points. NOTE: this requirement is not represented on the sample output below)**

Print to the screen: “Thank you for visiting the <YYY> store” where <YYY> is your last name.

//\*\*\* Your Name Here \*\*\* #include <iostream> #include<iomanip> #include <fstream>

struct Inventory

{

char desc[30]; int qty;

double wholeSaleCost;

};

// Function prototypes void addRecord(fstream &);

void viewRecord(fstream &); void changeRecord(fstream &);

int main()

{

using namespace std;

fstream inventoryFile; int choice;

cout << setprecision(2) << fixed;

do

{

// Display the menu.

cout << "\n1. Add a new record\n";

cout << "2. View an existing record by record number\n"; cout << "3. Change an existing record\n";

cout << "4. Exit\n\n"; do

{

cout << "Enter your choice (1-4): "; cin >> choice;

}

while (choice < 1 || choice > 4);

// Process the selection. switch (choice)

{

// Choice 1 is to add a record. case 1:

addRecord(inventoryFile); break;

// Choice 2 is to view a record. case 2:

viewRecord(inventoryFile); break;

// Choice 3 is to change a record. case 3:

changeRecord(inventoryFile);

}

} while (choice != 4);

system ("pause"); return 0;

}

# Hints:

1. Refer to textbook example progams 12-13 through 12-22.
2. You should manually delete the inventory test file every time you run your program.

If you don’t delete the inventory file and run your program again, you may start working with a file that is already in existence as opposed to creating a new file.

1. Note – students have asked: Why was the string class not included? This prevents the use of getline() and substring methods that could have made the code more organized and efficient.

The main reason is: it is easier to send a character array to a file because it is fixed in size.

**Sample Output**



