**Name:**

**Advanced Programming in C++**

**Lab Exercise 4/13/2020**

In this exercise you will use files to store and retrieve data. When you complete the exercise, you are to print out your source code and attach it to this sheet as well as a sample output.

Write a program that uses a structure to store the inventory information in file:

Item Description

Part Number

Quantity on Hand

Wholesale cost

Retail cost

Date added to inventory

The program should have a menu that allows the user to perform the following tasks:

Add new records to a file

Display any record in the file

Change any record in a file

In this project, data is stored in a file. We will use a struct as the container for the data. The main program will use a vector of structs as the main data structure for the program. The program will contain 6 functions to perform all program tasks. Here are the prototypes for those functions.

void printRecords(vector<Inventory>);

void addRecord(vector<Inventory> &); //reference parameter

void loadRecords(vector<Inventory> &); //reference parameter

void editRecord(vector<Inventory> &); //reference parameter

void displayRecord(vector<Inventory>);

void update(vector<Inventory>);

Note: Functions that modify the vector are passed by reference.

Here is the structure definition

//Data structure definition

struct Inventory

{

string description;

string partNumber;

int quantity;

double wholesale;

double retail;

string date;

//Structure constructor

Inventory(string d, string pn, int q, double ws, double ret,

string dt)

{

description = d;

partNumber = pn;

quantity = q;

wholesale = ws;

retail = ret;

date = dt;

}

};

Note: it contains not only the structure member but a constructor to initialize all structure members to a specified value.

1. Add the following code to the printRecords function.

int length = ww.size();

for (int i = 0; i < length; i++)

{

cout << "Record " << i+1 << endl;

cout << "Description: " << ww[i].description << endl;

cout << "Part number: " << ww[i].partNumber << endl;

cout << "Quantity: " << ww[i].quantity << endl;

cout << "Wholesale price: " << ww[i].wholesale << endl;

cout << "Retail price: " << ww[i].retail << endl;

cout << "Aquisition date: " << ww[i].date << endl;

cout << endl;

}

1. Add the following code to the addRecord function.

string d, pn, dt;

int q;

double ws, ret;

cout << "Ready to add a record (Hit key to continue)";

cin.ignore();

cout << "Enter part description: ";

getline(cin, d);

cout << "Part number: ";

cin >> pn;

cout << "Quantity: ";

cin >> q;

cout << "Wholesale price: ";

cin >> ws;

cout << "Retail price: ";

cin >> ret;

cout << "Date aquired (mm/dd/yyyy): ";

cin >> dt;

cin.ignore();

ww.push\_back(Inventory(d,pn,q,ws,ret,dt)); //add new record

update(ww);

1. Add the following code to the loadRecords function.

string d, pn, dt;

int q;

double ws, ret;

ifstream infile;

infile.open("wally.txt");

while (!infile.eof())

{

getline(infile, d);

infile >> pn;

infile >> q;

infile >> ws;

infile >> ret;

infile >> dt;

infile.ignore();

ww.push\_back(Inventory(d,pn,q,ws,ret,dt)); //construct record

}

infile.close();

1. Add the following code to the editRecord function.

int length = ww.size();

int recordNumber;

cout << "Enter record to edit (1 - " << length << "): ";

cin >> recordNumber;

cin.ignore();

cout << "Enter part description: ";

getline(cin, ww[recordNumber - 1].description);

cout << "Part number: ";

cin >> ww[recordNumber - 1].partNumber;

cout << "Quantity: ";

cin >> ww[recordNumber - 1].quantity;

cout << "Wholesale price: ";

cin >> ww[recordNumber - 1].wholesale;

cout << "Retail price: ";

cin >> ww[recordNumber - 1].retail;

cout << "Date aquired (mm/dd/yyyy): ";

cin >> ww[recordNumber - 1].date;

cin.ignore();

cout << endl << endl;

update(ww);

1. Add the following code to the displayRecord function.

int length = ww.size();

int recordNumber;

cout << "Enter record to display (1 - " << length << "): ";

cin >> recordNumber;

cout << "Record " << recordNumber << endl;

cout << "Description: " << ww[recordNumber - 1].description << endl;

cout << "Part number: " << ww[recordNumber - 1].partNumber << endl;

cout << "Quantity: " << ww[recordNumber - 1].quantity << endl;

cout << "Wholesale price: " << ww[recordNumber - 1].wholesale

<< endl;

cout << "Retail price: " << ww[recordNumber - 1].retail << endl;

cout << "Aquisition date: " << ww[recordNumber - 1].date << endl;

cout << endl;

1. Add the following code to the update function.

ofstream outfile;

int length = ww.size();

int i;

outfile.open("wally.txt");

//write all records except the last

for (i = 0; i < length - 1; i++)

{

outfile << ww[i].description << endl;

outfile << ww[i].partNumber << endl;

outfile << ww[i].quantity << endl;

outfile << ww[i].wholesale << endl;

outfile << ww[i].retail << endl;

outfile << ww[i].date << endl;

}

//write the last record

outfile << ww[i].description << endl;

outfile << ww[i].partNumber << endl;

outfile << ww[i].quantity << endl;

outfile << ww[i].wholesale << endl;

outfile << ww[i].retail << endl;

outfile << ww[i].date; //no endl at end of file

outfile.close();

1. Run and test your program. When you have it working copy the output to a word processing document and turn in.