SOURCE CODE

#include <iostream>

#include <iomanip>

#include <string>

#include <vector>

#include <windows.h>

#include<bits/stdc++.h>

#include <fstream>

#include <sstream>

#include <ctype.h>

using namespace std;

main()

{

system("color 64");

//Declare variable with suitable data type

string choicemenu;

bool mainOn = true;

string selectItemMenu;

bool itemOn = true;

string selectManualMenu;

bool manualOn = true;

string selectSellMenu;

bool sellOn = true;

string searchMenuSelect;

bool searchOn = true;

string printMenuSelect;

bool printOn = true;

int max = 999;

vector<int> itemID(max),pOrdered(max),pInStore(max),pSold(max);

vector<double> manufPrice(max), sellingPrice(max), totalSellingPrice(max);

vector<string> itemName(max);

int id,order,store,sold;

double mPrice, sPrice;

string name;

int idNew,orderNew,storeNew,soldNew;

double mPriceNew, sPriceNew;

string nameNew;

int totalInStore;

double totalPrice;

int i,j,check = 0, currIndex = 0, tempSell = 0;

totalSellingPrice.push\_back(0);

int itemIDAccept;

bool idCheck;

int p;

string updateMenuSelect;

itemID.push\_back(0);

//this loop will keep repeating until the user exit the program

while(mainOn != false)

{

mainOn = true, itemOn = true, printOn = true, sellOn = true, check = 0;

//menu

cout << endl;

cout << " Insert input" << endl ;

cout << " 1. Insert item" << endl ;

cout << " 2. Sell item" << endl ;

cout << " 3. Check item" << endl ;

cout << " 4. Print " << endl ;

cout << " 5. End program" << endl ;

cout << " Enter: " ;

cin >> choicemenu ;

//if user choose 1(Insert item)

if(choicemenu == "1")

{

//beep sound

Beep(245, 456);

while(itemOn!=false)

{

cout << endl;

cout << endl;

cout << "\*\* Method of entering data \*\*" << endl;

cout << " 1.Manual" << endl;

cout << " 2.Source File ( File name required)"<< endl;

cout << " 3.Back to Main Menu"<< endl;

//ask user to select a number based on the menu display

cout << endl << "Please select a number from the enter item menu and press return : ";

cin >> selectItemMenu;

if (selectItemMenu == "1")

{

//Intiliaze i to currIndex for each time the menu is selected/restart (LCV)

i = currIndex;

manualOn = true;

//The menu will repeat until itemOn = true and i > max / user exit the menu

while (manualOn != false)

{

{

idCheck = false;

//user enter item id

cout << "ITEM ID >> ";

cin >> id;

for(int z=0;z<currIndex;z++)

{

if(id==itemID[z])

{

idCheck = true;

p = z;

}

}

if(idCheck != true)

{

itemID[i] = id;

cin.ignore();

//user enter the data including item name, pieces ordered, manufacture price and selling price

cout << "ITEM NAME >> ";

getline(cin,name);

itemName[i] = name;

cout << "PIECES ORDERED >> ";

cin >> order;

pOrdered[i] = order;

cout << "MANUFACTURES PRICE >> ";

cin >> mPrice;

manufPrice[i] = mPrice;

cout << "SELLING PRICE >> ";

cin >> sPrice;

sellingPrice[i] = sPrice;

//the new item added will be initialized with this value

pInStore[i] = order;

pSold[i] = 0;

if(cin.fail())

{

cin.clear();

cin.ignore(numeric\_limits<std::streamsize>::max(), '\n');

//invalid input will be erased from the database

itemID.erase(itemID.begin() + currIndex);

itemName.erase(itemName.begin() + currIndex);

pOrdered.erase(pOrdered.begin() + currIndex);

pInStore.erase(pInStore.begin() + currIndex);

manufPrice.erase(manufPrice.begin() + currIndex);

sellingPrice.erase(sellingPrice.begin() + currIndex);

pSold.erase(pSold.begin() + currIndex);

//the output, if invalid input entered

cout << "ERROR 110 >> YOUR INPUT IS INVALID" << endl;

}

else

{

//calculate total item

totalInStore += pInStore[currIndex];

//calculate total price of item

totalSellingPrice[currIndex] = sellingPrice[currIndex]\*pInStore[currIndex];

totalPrice += totalSellingPrice[currIndex];

//update currIndex

currIndex++;

i++;

}

}

else

{

cout << "ITEM ID: " << id << " overlapped" << endl;

cout << "Rewrite ID? yes[y] / no[n] >> ";

cin >> updateMenuSelect;

if( updateMenuSelect == "y" || updateMenuSelect == "Y" )

{

nameNew = itemName[p];

orderNew = pOrdered[p];

mPriceNew = manufPrice[p];

sPriceNew = sellingPrice[p];

storeNew = pInStore[p];

soldNew = pSold[p];

//calculate total item

totalInStore -= pInStore[p];

//calculate total price of the item

totalPrice -= totalSellingPrice[p];

cin.ignore();

//user enter the data

cout << " NEW ITEM NAME >> ";

getline(cin,name);

itemName[p] = name;

cout << "NEW PIECES ORDERED >> ";

cin >> order;

pOrdered[p] = order;

cout << "NEW MANUFACTURES PRICE >> ";

cin >> mPrice;

manufPrice[p] = mPrice;

cout << "NEW SELLING PRICE >> ";

cin >> sPrice;

sellingPrice[p] = sPrice;

cout << "NEW NUMBER OF ITEM IN STORE >> ";

cin >> store;

pInStore[p] = store;

cout << "NEW NUMBER OF ITEM SOLD >> ";

cin >> sold;

pSold[p] = sold;

if(cin.fail())

{

cin.clear();

cin.ignore(numeric\_limits<std::streamsize>::max(), '\n');

itemName[p] = nameNew;

pOrdered[p] = orderNew;

manufPrice[p] = mPriceNew;

sellingPrice[p] = sPriceNew;

pInStore[p] = storeNew;

pSold[p] = soldNew;

//calculate total of item

totalInStore += pInStore[p];

//calculate total price of all item

totalSellingPrice[p] = sellingPrice[p]\*pInStore[p];

totalPrice += totalSellingPrice[p];

//the output, if invalid input entered

cout << "ERRIR 111: INPUT IS INVALID" << endl;

}

else

{

//calculate total of item in the store

totalInStore += pInStore[p];

//calculate total price of all item

totalSellingPrice[p] = sellingPrice[p]\*pInStore[p];

totalPrice += totalSellingPrice[p];

cout << "Item ID " << id << " updated" << endl;

}

}

else if(updateMenuSelect != "N")

{

cout << "[ERROR112 : INVALID INPUT]" << endl;

}

}

int holdID,holdOrder,holdInStore,holdSold;

double holdmPrice,holdsPrice;

string holdName;

//Do a sorting arrays in ascending order based on Item's name

for(i=1;i<currIndex;i++)

{

for(j=0;j<currIndex-i;j++)

{

if(itemName[j]>itemName[j+1])

{

holdID = itemID[j];

itemID[j] = itemID[j+1];

itemID[j+1]=holdID;

holdName = itemName[j];

itemName[j] = itemName[j+1];

itemName[j+1]=holdName;

holdOrder = pOrdered[j];

pOrdered[j] = pOrdered[j+1];

pOrdered[j+1]=holdOrder;

holdInStore = pInStore[j];

pInStore[j] = pInStore[j+1];

pInStore[j+1]=holdInStore;

holdSold = pSold[j];

pSold[j] = pSold[j+1];

pSold[j+1]=holdSold;

holdmPrice = manufPrice[j];

manufPrice[j] = manufPrice[j+1];

manufPrice[j+1]=holdmPrice;

holdsPrice = sellingPrice[j];

sellingPrice[j] = sellingPrice[j+1];

sellingPrice[j+1]=holdsPrice;

}

}

}

cout << "Add another item? yes[y] / no[n] >> ";

cin >> selectManualMenu;

{

switch(selectManualMenu.at(0))

{

case 'y':

manualOn = true;

break;

case 'n':

manualOn = false;

break;

default:

cout << "ERROR 113: PLEASE TRY AGAIN" << endl << endl;

manualOn = false;

}

}

}

}

}

else if (selectItemMenu == "2")

{

//Beep sound

Beep(503,561);

string idStr,orderStr,storeStr,soldStr, mPriceStr, sPriceStr;

i = currIndex;

string infilename;

cout << "NAME/LOCATION OF THE FILE >> " << endl;

cin >> infilename;

ifstream inFile;

inFile.open(infilename.c\_str());

if(inFile.is\_open())

{

cout << "Searching for file..." << endl;

string line;

getline(inFile, line);

while(!inFile.eof() && !inFile.fail())

{

idCheck = false;

getline(inFile,idStr,',');

stringstream idNum(idStr);

idNum >> id;

for(int z=0;z<currIndex;z++)

{

if(id==itemID[z])

{

idCheck = true;

p=z;

}

}

if(idCheck != true)

{

itemID[i] = id;

getline(inFile,name, ',');

itemName[i] = name;

getline(inFile,orderStr, ',');

stringstream orderNum(orderStr);

orderNum >> order;

pOrdered[i] = order;

getline(inFile,storeStr, ',');

stringstream storeNum(storeStr);

storeNum >> store;

pInStore[i] = store;

getline(inFile,soldStr, ',');

stringstream soldNum(soldStr);

soldNum >> sold;

pSold[i] = sold;

getline(inFile,mPriceStr, ',');

stringstream mPriceNum(mPriceStr);

mPriceNum >> mPrice;

manufPrice[i] = mPrice;

getline(inFile,sPriceStr, '\n');

stringstream sPriceNum(sPriceStr);

sPriceNum >> sPrice;

sellingPrice[i] = sPrice;

//print total item

totalInStore += pInStore[currIndex];

//total price

totalSellingPrice[currIndex] = sellingPrice[currIndex]\*pInStore[currIndex];

totalPrice += totalSellingPrice[currIndex];

currIndex++;

i++;

}

else

{

totalInStore -= pInStore[p];

totalPrice -= totalSellingPrice[p];

getline(inFile,name, ',');

itemName[p] = name;

getline(inFile,orderStr, ',');

stringstream orderNum(orderStr);

orderNum >> order;

pOrdered[p] = order;

getline(inFile,storeStr, ',');

stringstream storeNum(storeStr);

storeNum >> store;

pInStore[p] = store;

getline(inFile,soldStr, ',');

stringstream soldNum(soldStr);

soldNum >> sold;

pSold[p] = sold;

getline(inFile,mPriceStr, ',');

stringstream mPriceNum(mPriceStr);

mPriceNum >> mPrice;

manufPrice[p] = mPrice;

getline(inFile,sPriceStr, '\n');

stringstream sPriceNum(sPriceStr);

sPriceNum >> sPrice;

sellingPrice[p] = sPrice;

//print total of item

totalInStore += pInStore[p];

//total price

totalSellingPrice[p] = sellingPrice[p]\*pInStore[p];

totalPrice += totalSellingPrice[p];

}

}

int holdID,holdOrder,holdInStore,holdSold;

double holdmPrice,holdsPrice;

string holdName;

//Do a sorting arrays in ascending order based on Item's name

for(i=1;i<currIndex;i++)

{

for(j=0;j<currIndex-i;j++)

{

if(itemName[j]>itemName[j+1])

{

holdID = itemID[j];

itemID[j] = itemID[j+1];

itemID[j+1]=holdID;

holdName = itemName[j];

itemName[j] = itemName[j+1];

itemName[j+1]=holdName;

holdOrder = pOrdered[j];

pOrdered[j] = pOrdered[j+1];

pOrdered[j+1]=holdOrder;

holdInStore = pInStore[j];

pInStore[j] = pInStore[j+1];

pInStore[j+1]=holdInStore;

holdSold = pSold[j];

pSold[j] = pSold[j+1];

pSold[j+1]=holdSold;

holdmPrice = manufPrice[j];

manufPrice[j] = manufPrice[j+1];

manufPrice[j+1]=holdmPrice;

holdsPrice = sellingPrice[j];

sellingPrice[j] = sellingPrice[j+1];

sellingPrice[j+1]=holdsPrice;

}

}

}

cout << "The program is inserting data from your file..." << endl;

inFile.close();

}

else

{

cout << "ERROR 115: File is unable to open" << endl;

}

}

else if (selectItemMenu == "3")

{

system("color 46"); // white background and black text colour

itemOn = false;

}

else

{

cout << "ERROR 116: " << endl << endl;

}

}

}

//If user choose 2(Sell item)

else if(choicemenu == "2")

{

//Beep sound

Beep(546, 519);

system("color 5B");

while(sellOn != false)

{

//User enter the item ID that already sold

cout << endl << "Please enter item ID : ";

cin >> itemIDAccept;

check = 0;

if(cin.fail())

{

cin.clear();

cin.ignore(numeric\_limits<std::streamsize>::max(), '\n');

//the ouput, if invalid input is entered

cout << "ERROR 117: Invalid Input" << endl;

}

else

{

for (i = 0; i < currIndex; i++)

{

if (itemIDAccept == itemID[i])

{

//User enter number of pieces sold

cout << "NUMBER OF PIECES SOLD : ";

cin >> tempSell;

pSold[i] += tempSell;

//Update total item in store

pInStore[i] -= tempSell;

//Update total selling price

totalSellingPrice[i] -= (tempSell \* sellingPrice[i]);

totalInStore -= tempSell;

totalPrice -= (tempSell \* sellingPrice[i]);

check = 1;

break;

}

}

}

if (check == 1)

{

//Ask the user

cout << "Add another sold item? yes[y] / no[n] >> ";

cin >> selectSellMenu;

{

//If y is entered, the process will be repeated

switch(selectSellMenu.at(0))

{

case 'y':

sellOn = true;

break;

case 'n':

sellOn = false;

break;

default:

cout << "ERROR 119; INVALID INPUT" << endl << endl;

sellOn = false;

}

}

}

else

{

cout << itemIDAccept << " << ITEM is not available in the program" << endl;

cout << "Enter new item by choosing 1 in the main menu" << endl;

break;

}

}

}

//If user choose 3(Check item)

else if(choicemenu == "3")

{

//Beep sound

Beep(505, 478);

system("color 6B");

while(searchOn != false)

{

//Ask user to enter Item ID to check the avalability of the item

cout << endl << "ITEM ID >> ";

cin >> itemIDAccept;

check = 0;

if(cin.fail())

{

cin.clear();

cin.ignore(numeric\_limits<std::streamsize>::max(), '\n');

//the output, if invalid input is entered

cout << "ERROR 120: PLEASE TRY AGAIN" << endl;

}

else

{

for (i = 0; i < currIndex; i++)

{

if (itemIDAccept == itemID[i])

{

cout << "Item " << itemIDAccept << " is at index " << i << endl;

cout << "Total item " << itemIDAccept << " in the store: " << pInStore[i] << " pieces" << endl;

check = 1;

break;

}

}

}

if (check == 1)

{

//Ask the user

cout << "Check another item? yes[y] / no[n] >> ";

cin >> searchMenuSelect;

{

//If y is entered, the user has to enter the Item ID

switch(searchMenuSelect.at(0))

{

case 'y':

searchOn = true;

break;

case 'n':

searchOn = false;

break;

default:

cout << "TRY AGAIN" << endl << endl;

searchOn = false;

}

}

}

else

{

cout << itemIDAccept << " item choosen isn't available in the program'" << endl;

cout << "Re-enter 1 in the main menu to enter new item" << endl;

cout << endl;

break;

}

}

}

//If user choose 4(Print)

else if(choicemenu == "4")

{

//Beep sound

Beep(574, 505);

while(printOn != false)

{

system("color 7D");

cout<< left << setw(15) << "itemID" << setw(20) << "itemName" << setw(20) << "pOrdered" << setw(15) << "pInStore";

cout<< left << setw(15) << "pSold" << setw(15) << "manufPrice" << setw(15) << "sellingPrice" << endl;

//This program will print out all the data

for(i=0;i<currIndex;i++)

{

cout << left << setw(15)<< itemID[i] << setw(20) << itemName[i] << setw(20) << pOrdered[i] << setw(15) << pInStore[i];

cout << left << setw(15) << pSold[i] << setw(15) << setprecision(2) << fixed << manufPrice[i] << setw(15) << setprecision(2) << fixed << sellingPrice[i] << endl;

}

cout << endl << "Total Inventory : $" << setprecision(2) << fixed << totalPrice << endl;

cout << "Total number of item in the store: " << totalInStore << endl;

string printFileSelect;

cout << endl;

//Ask the user if they want to create a file

cout << "Create a file for this report?" << endl;

cout << "1. Yes (Full report)" << endl;

cout << "2. Yes (Source file)" << endl;

cout << "3. No" << endl;

cout << endl << "Enter item number in the menu and press return >> ";

cin >> printFileSelect;

string outfilename;

switch(printFileSelect.at(0))

{

case '1':

{

ofstream outFile;

cout << "File name? (example: filename.txt) >> " << endl;

cin >> outfilename;

outFile.open(outfilename.c\_str());

if(outFile.is\_open())

{

outFile << left << setw(15)<< "itemID" << setw(20) << "itemName" << setw(20) << "pOrdered" << setw(15) << "pInStore";

outFile << left << setw(15) << "pSold" << setw(15) << "manufPrice" << setw(15) << "sellingPrice" << endl;

//This program will print out all the data

for(i=0;i<currIndex;i++)

{

outFile << left<< setw(15)<< itemID[i] << setw(20) << itemName[i] << setw(20) << pOrdered[i] << setw(15) << pInStore[i];

outFile << left<< setw(15) << pSold[i] << setw(15) << setprecision(2) << fixed << manufPrice[i] << setw(15) << setprecision(2) << fixed << sellingPrice[i] << endl;

}

outFile << endl << "Total Inventory >> RM" << setprecision(2) << fixed << totalPrice << endl;

outFile << "Total number of item >> " << totalInStore << endl;

outFile.close();

cout << "File have been created within the same location as your program." << endl;

}

else

{

cout << "\*\*\*\*\*\*\*\*\*\*\*Unable to create file\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

}

}

break;

case '2':

{

ofstream outFile;

Beep(245,645);

cout << "File name? (example: filename.txt) >> " << endl;

cin >> outfilename;

outFile.open(outfilename.c\_str());

if(outFile.is\_open())

{

cout << "Creating file..." << endl;

outFile << "itemID" << " " << "itemName" << " " << "pOrdered" << " " << "pInStore";

outFile << " " << "pSold" << " " << "manufPrice" << " " << "sellingPrice" << endl;

//This program will print out all the data

for(i=0;i<currIndex;i++)

{

outFile << itemID[i] << "," << itemName[i] << "," << pOrdered[i] << "," << pInStore[i];

outFile << "," << pSold[i] << "," << setprecision(2) << fixed << manufPrice[i] << "," << setprecision(2) << fixed << sellingPrice[i] << endl;

}

outFile.close();

cout << "File created succesfully" << endl;

}

else

{

cout << "There seems like a problem in creating a file :/" << endl;

}

}

break;

case '3':

Beep(545,645);

printOn = true;

break;

default:

cout << "ERROR DETECTED IN INPUT" << endl << endl;

printOn = false;

}

cout << endl << "Print another report? yes[y] / no[n] >> ";

cin >> printMenuSelect;

switch(printMenuSelect.at(0))

{

case 'y':

printOn = true;

break;

case 'n':

system("color 4A");

printOn = false;

break;

default:

cout << "ERROR 121: INPUT IS INVALID" << endl << endl;

printOn = false;

}

}

//If user choose 5(End Program)

}

else if(choicemenu == "5")

{

//Beep sound

Beep(1568, 200);

Beep(1245, 500);

mainOn = false;

}

else

{

cout << "ERROR IN INPUT" << endl << endl;

}