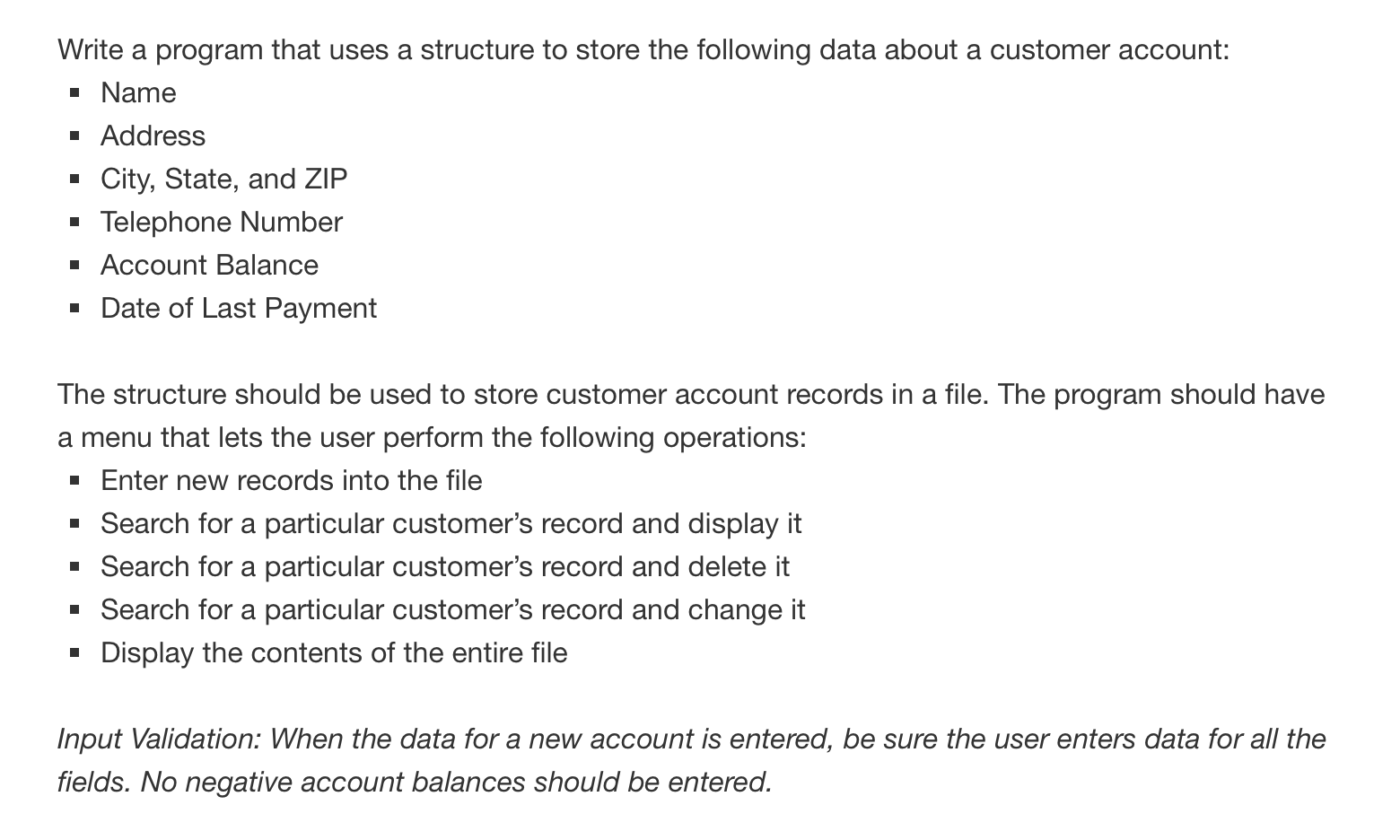
COMSC 165 PROJECT DUE: JULY-26 11:59 PM



#ifndef CUSTOMERACCOUNTDATA\_H

#define CUSTOMERACCOUNTDATA\_H

const int SIZE = 80;

struct CustomerAccount{

// customer name

char name[SIZE];

// customer address

char address[SIZE];

// customer city

char city[SIZE];

// customer state

char state[SIZE];

// customer zip code

char zipCode[SIZE];

// customer phone number

char phoneNumber[SIZE];

// customer account balance

char balance[SIZE];

// customer last payment date

char lastPaymentDate[SIZE];

};

#endif // CUSTOMERACCOUNTDATA\_H

#ifndef CUSTOMERACCOUNTPROGRAM\_H

#define CUSTOMERACCOUNTPROGRAM\_H

class CustomerAccountProgram{

public:

// displays a menu for the user to interact with an item

int inventoryMenu();

// add new records to the file

void addNewRecords();

// display any record in the file

void displayRecords();

// change any record in the file

void changeRecords();

};

#endif // CUSTOMERACCOUNTPROGRAM\_H

#include <iostream>

#include <fstream>

#include <cstdlib>

#include "CustomerAccountProgram.h"

#include "CustomerAccount.h"

using namespace std;

// for the user to perform tasks

int CustomerAccountProgram::inventoryMenu(){

int choice;

cout << "MENU" << endl;

cout << "1: Add new items to the inventory record." << endl;

cout << "2: Display inventory item." << endl;

cout << "3: Change any item in the inventory record." << endl;

cout << "4: Quit the program.\n" << endl;

cin >> choice;

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

cout << "You entered an invalid option. Please try again." << endl;

cin >> choice;

}

return choice;

}

// add new records to the file

void CustomerAccountProgram::addNewRecords(){

// create fstream object

// open the file for output in binary mode

fstream fileObject;

fileObject.open("customerAccount.txt", ios::out);

if (!fileObject.is\_open()){

cout << "Error reading file!" << endl;

exit(EXIT\_FAILURE);

}else{

// create Customer Account object

CustomerAccount caObject;

char repeat;

int i = 0;

do{

// get user input for customer account information

cout << "Provide the Customer Account Information." << endl;

cout << "\nCustomer Name: ";

cin.ignore();

cin.getline(caObject.name, SIZE);

cout << "Address Line 1: ";

cin.ignore();

cin.getline(caObject.address, SIZE);

cout << "City: ";

cin.ignore();

cin.getline(caObject.city, SIZE);

cout << "State: ";

cin.ignore();

cin.getline(caObject.state, SIZE);

cout << "Zip: ";

cin.ignore();

cin.getline(caObject.zipCode, SIZE);

cout << "Phone Number: ";

cin.ignore();

cin.getline(caObject.phoneNumber, SIZE);

cout << "Account Balance: ";

cin.ignore();

cin.getline(caObject.balance, SIZE);

while(caObject.balance < 0){

cout << "Invalid entry. Please enter a positve balance: ";

cin >> caObject.balance;

}

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

cin.ignore();

cin.getline(caObject.lastPaymentDate, SIZE);

// write cusomter account data to file

fileObject.write(reinterpret\_cast<char\*>(&caObject), sizeof(caObject));

cout << "Do you want to add another item? (y/n) ";

cin >> repeat;

}while(repeat == 'y');

cout << endl;

// close the file

fileObject.close();

}

}

// display any record in the file

void CustomerAccountProgram::displayRecords(){

fstream fileObject;

fileObject.open("customerAccount.txt", ios::in);

CustomerAccount caObject;

fileObject.read(reinterpret\_cast<char\*>(&caObject), sizeof(caObject));

while(!fileObject.eof()){

cout << "\nCustomer Account Name: ";

cout << caObject.name << endl;

cout << "Address: ";

cout << caObject.address << " "

<< caObject.city << " "

<< caObject.state << " "

<< caObject.zipCode << endl;

cout << "Phone Number: ";

cout << caObject.phoneNumber << endl;;

cout << "Account Balance: ";

cout << caObject.balance << endl;

cout << "Last Payment Date: ";

cout << caObject.lastPaymentDate << endl;

fileObject.read(reinterpret\_cast<char\*>(&caObject), sizeof(caObject));

}

// close the file

fileObject.close();

}

// change any record in the file

void CustomerAccountProgram::changeRecords(){

fstream fileObject;

fileObject.open("customerAccount.txt", ios::in | ios::out);

CustomerAccount caObject;

int recordNumber = 0;

// move to desired item and display

cout << "Enter the item # you'd like to edit: ";

cin >> recordNumber;

fileObject.seekg((recordNumber)\*sizeof(caObject), ios::beg);

fileObject.read(reinterpret\_cast<char\*>(&caObject), sizeof(caObject));

cout << "Old record: " << endl;

cout << "Customer Account Name: ";

cout << caObject.name << endl;

cout << "Address: ";

cout << caObject.address << " "

<< caObject.city << " "

<< caObject.state << " "

<< caObject.zipCode << endl;

cout << "Phone Number: ";

cout << caObject.phoneNumber << endl;;

cout << "Account Balance: ";

cout << caObject.balance << endl;

cout << "Last Payment Date: ";

cout << caObject.lastPaymentDate << endl;

cout << "Enter new record: " << endl;

// get user input for customer account information

cout << "Provide the Customer Account Information." << endl;

cout << "\nCustomer Name: ";

cin.ignore();

cin.getline(caObject.name, SIZE);

cout << "Address Line 1: ";

cin.ignore();

cin.getline(caObject.address, SIZE);

cout << "City: ";

cin.ignore();

cin.getline(caObject.city, SIZE);

cout << "State: ";

cin.ignore();

cin.getline(caObject.state, SIZE);

cout << "Zip: ";

cin.ignore();

cin.getline(caObject.zipCode, SIZE);

cout << "Phone Number: ";

cin.ignore();

cin.getline(caObject.phoneNumber, SIZE);

cout << "Account Balance: ";

cin.ignore();

cin.getline(caObject.balance, SIZE);

while(caObject.balance < 0){

cout << "Invalid entry. Please enter a positve balance: ";

cin >> caObject.balance;

}

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

cin.ignore();

cin.getline(caObject.lastPaymentDate, SIZE);

fileObject.seekg((recordNumber)\*sizeof(caObject), ios::beg);

fileObject.write(reinterpret\_cast<char\*>(&caObject), sizeof(caObject));

// close the file

fileObject.close();

}

#include <iostream>

#include <fstream>

#include "CustomerAccountProgram.h"

#include "CustomerAccount.h"

using namespace std;

int main(){

CustomerAccountProgram customerAccount;

int choice;

choice = customerAccount.inventoryMenu();

while(choice != 4){

switch(choice){

case 1:{

customerAccount.addNewRecords();

break;

}

case 2:{

customerAccount.displayRecords();

break;

}

case 3:{

customerAccount.changeRecords();

break;

}

}

choice = customerAccount.inventoryMenu();

}

return 0;

}

