Data Structure and Algorithms

Lab Journal - Lab 0

Name: _____ABDULLAH_____

Enrollment #: _____01-134232-

013_____

Class/Section: ___BSCS 3D_____

Objective

This lab is intended to provide a recap of concepts in C++ and OOP that will be frequently used in Data Structure and Algorithms course.

Task 1: Exercises

Implement the following exercises.

Exercise 1 :

a) Declare a class named **House** for a real estate locator service. The following information should be included:

Owner: (a string of up to 20 characters)
Address: (a string of up to 20 characters)
Bedrooms: (an integer)
Price (floating point)

b) Declare **available** to be an array of 100 objects of class **House.**

c) Write a function to read values into the members of an object of **House**.

d) Write a driver program to test the data structures and the functions you have developed.

The driver program should read in house entries into the **available** array. After the code for entering the data, you should write code to output the data that you have entered to verify that it is correct.

Your program should look like this:

Enter Owner : *M. Khan*
Enter Address : *G-9, Islamabad*
Number of Bedrooms? : *4*

Price : *4500000*
Enter another house? N The output should look like:

Owner	Address	Bedrooms	Price
M. Khan	G-9, Islamabad	4	4500000

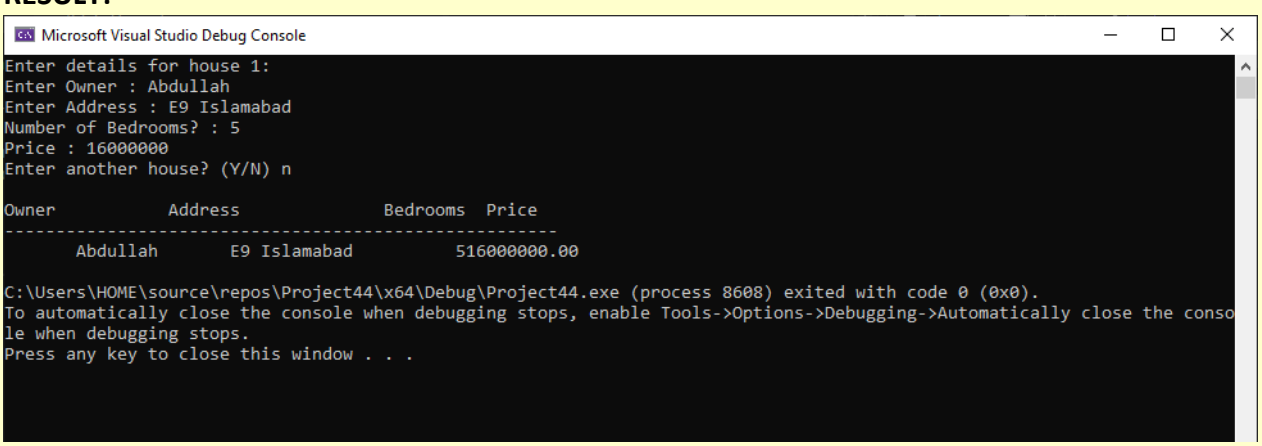
Extra Credit:
The real estate company is very happy with the program that was developed in the earlier to track their listings. Now they want to add some features to the processing. Additional features:

Search for a house that meets a potential buyer's specifications for the following:

- The price is not more than a specified amount
- The size is not less than a specified number of bedrooms The house with lowest price
- The largest house (with maximum number of bedrooms)
- In a given city
- With best ratio price/size
- The user may enter a "?" to indicate no preference.

Print all the entries that meet the buyer's need.

RESULT:



CODE:

House.h:

```
#include <string>

using namespace std;

class House {
public:
    string owner;
```

```
    string address;
    int bedrooms;
    float price;


    House();

    void readHouse();

    void displayHouse() const;
};
House.cpp:
#include "House.h"
#include <iostream>
#include <iomanip>

using namespace std;

House::House() : owner(""), address(""), bedrooms(0), price(0.0f) {}


void House::readHouse() {
    cout << "Enter Owner : ";
    getline(cin, owner);
    cout << "Enter Address : ";
    getline(cin, address);
    cout << "Number of Bedrooms? : ";
    cin >> bedrooms;
    cout << "Price : ";
    cin >> price;
    cin.ignore();
}


void House::displayHouse() const {
    cout << setw(15) << owner
        << setw(20) << address
        << setw(10) << bedrooms
        << setw(10) << fixed << setprecision(2) << price << endl;
}
```

Source.cpp

```
#include <iostream>
#include "House.h"

using namespace std;

int main() {
    const int MAX_HOUSES = 100;
    House available[MAX_HOUSES];
    int numHouses = 0;
    char choice;

    do {
        if (numHouses >= MAX_HOUSES) {
            cout << "Cannot add more houses. Maximum limit reached." << endl;
            break;
        }
```

```
        cout << "Enter details for house " << (numHouses + 1) << ":" << endl;
        available[numHouses].readHouse();
        numHouses++;

        cout << "Enter another house? (Y/N) ";
        cin >> choice;
        cin.ignore();
    } while (choice == 'Y' || choice == 'y');

    cout << "\nOwner           Address              Bedrooms  Price" << endl;
    cout << "-------------------------------------------------------" << endl;

    for (int i = 0; i < numHouses; ++i) {
        available[i].displayHouse();
    }

    return 0;
}
```

Exercise 2:

Assume that a file contains the midterm1, midterm2 and final exam scores and names of students of a class. Write a C++ program to read the input file and produce an output file containing the original and average scores for each student. Suppose that the weights of the exams are as follows:

midterm1 – 25%
midterm2 – 25% final
– 50%.

The average score of a student is calculated using the formula:

$0.25*MT1 + 0.25*MT2 + 0.5*FIN$
Result

```
John Doe
85 90 95
Jane Smith
78 82 88
```

```
Name                 Midterm1  Midterm2   Final   Average
-------------------------------------------------------
John Doe               85.00     90.00    95.00    92.50
Jane Smith             78.00     82.00    88.00    84.50
```

Code:

Student.h:

```
#pragma once
#include <string>

using namespace std;

class Student {
public:
    string name;
    float midterm1;
    float midterm2;
    float finalExam;
    float averageScore;


    Student() {};

    void calculateAverage();


    void display() const;
};
```

Student.cpp:

```
#include "Student.h"
#include <iostream>
#include <iomanip>

using namespace std;

// Constructor
```

```
Student::Student() : name(""), midterm1(0.0f), midterm2(0.0f), finalExam(0.0f),
averageScore(0.0f) {}

void Student::calculateAverage() {
    averageScore = 0.25f * midterm1 + 0.25f * midterm2 + 0.50f * finalExam;
}

void Student::display() const {
    cout << setw(20) << name
        << setw(10) << fixed << setprecision(2) << midterm1
        << setw(10) << fixed << setprecision(2) << midterm2
        << setw(10) << fixed << setprecision(2) << finalExam
        << setw(10) << fixed << setprecision(2) << averageScore << endl;
}
```

Main.cpp:

```
#include "Student.h"
#include <iostream>
#include <iomanip>

using namespace std;

Student::Student() : name(""), midterm1(0.0f), midterm2(0.0f), finalExam(0.0f),
averageScore(0.0f) {}


void Student::calculateAverage() {
    averageScore = 0.25f * midterm1 + 0.25f * midterm2 + 0.50f * finalExam;
}


void Student::display() const {
    cout << setw(20) << name
        << setw(10) << fixed << setprecision(2) << midterm1
        << setw(10) << fixed << setprecision(2) << midterm2
        << setw(10) << fixed << setprecision(2) << finalExam
        << setw(10) << fixed << setprecision(2) << averageScore << endl;
}
```

Exercise 3:

You will write a student grades "database" program. It will read data of students from a file and will let the user perform various operations on the data. You will have to store the student data in an array of objects.

Input:
The input file will look like:
4
3
Hassan Khan 99 87 90
Sara Nazir 90 98 99
Ali Zaidi 55 43 0
Raza Ahmad 100 100 100

That is:
number of students number of
grades (per student) Student name
grade grade ... grade
Student name grade grade ... grade

Data structure :

You will store all the information in an array of "student" objects. You may use the following class definition:

```
class student {
private:
    char name[30];
    int lab[10]; float
    average;
public:
     //Any functions you want to create
};
```

Result

```
Microsoft Visual Studio Debug Console
Enter the number of students: 2
Enter the number of grades per student: 2
Enter student 1 name: zaid
Enter 2 grades for zaid: 76 54
Enter student 2 name: subhan
Enter 2 grades for subhan: 88 91
Student Information:
Name: zaid
Grades:
76 54 Average: 65
Name: subhan
Grades:
88 91 Average: 89.5
```

Code:

Main.cpp:

```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

class student {
private:
    string name;
    int lab[10];
    float avg;
    int number_grades;

public:
    student(float a = 0.0, int g = 0) { avg = a; number_grades = g; }

    void setData(string student_name, int grades[], int count) {
        name = student_name;
        number_grades = count;
        for (int i = 0; i < count; i++) {
            lab[i] = grades[i];
        }
        calculate_average();
    }

    void calculate_average() {
        int sum = 0;
        for (int i = 0; i < number_grades; i++) {
            sum += lab[i];
        }
        avg = static_cast<float>(sum) / number_grades;
    }

    void display() const {
        cout << "Name: " << name << endl << "Grades: " <<endl;
        for (int i = 0; i < number_grades; i++) {
            cout << lab[i] << " ";
        }
        cout << "Average: " << avg <<endl;
    }

    void writetofile(ofstream& outFile) const {
        outFile << name;
        for (int i = 0; i < number_grades; i++) {
            outFile << lab[i] << " ";
        }
        outFile;
    }
};

const int A = 100;

int main() {

    int number_students, Grades;
```

```
    student students[A];

    cout << "Enter the number of students: ";
    cin >> number_students;
    cout << "Enter the number of grades per student: ";
    cin >> Grades;
    cin.ignore();

    for (int i = 0; i < number_students; i++) {
        string name;
        int grades[10];

        cout << "Enter student " << i + 1 << " name: ";
        getline(cin, name);

        cout << "Enter " << Grades << " grades for " << name << ": ";
        for (int j = 0; j < Grades; j++) {
            cin >> grades[j];
        }
        cin.ignore();

        students[i].setData(name, grades, Grades);
    }
    ofstream outputFile("students.txt");
    outputFile << number_students << Grades;
    for (int i = 0; i < number_students; i++) {
        students[i].writetofile(outputFile);
    }
    outputFile.close();
    cout << "Student Information:"<<endl;

    for (int i = 0; i < number_students; i++) {
        students[i].display();
    }

    return 0;
}
```

Implement the given exercises and get them checked by your instructor.

S No.	Exercise	Checked By:
1.	Exercise 1	
2.	Exercise 2	
3.	Exercise 3	

++++++++++++++++++++++++++