

Lab 9: [Experiments on reading and writing data to and from files]

Objective(s)

Learning how to use text data files for input and output sequentially in C++.

Tool(s)/Software

DevC++.

Description

- Getting input from the keyboard and sending output to the screen have several limitations.
- When dealing with a large amount of data, it may be more convenient to read inputs and produce outputs, to and from Files rather than manually typing in inputs and printing outputs to the screen. Data files also provide data persistence.

- **File I/O in C++ is a five-step process:**

1. Include the header file *fstream* in the program.

```
#include<fstream>
```

2. Declare file stream variables.

```
ifstream inFile;
```

```
ofstream outFile;
```

- The first statement declares inFile to be an input file stream variable.

- The second statement declares outFile to be an output file stream variable

3. Associate the file stream variables with the input/output sources or use open function.

```
ifstream inFile("inFile.txt");
```

```
ofstream outFile("outFile.txt");
```

4. Use the file stream variables with >>, << for reading or writing

```
inFile>> x;    outFile<<x;
```

5. Close the files.

```
inFile.close(); outFile.close();
```



Tasks/Assignments(s):

Task#1:

Write a C++ program that reads the input file *Scores.txt*. The file contains 3 values that represent the scores of a student in three courses. The program then prints those three marks on the screen and then calculates the average of these courses and prints the average **into the screen** and also **into the output file Average.txt**. (Note: if the delimiter-Character between numbers are spaces, then you do not need to use, the function: `cin.ignore()` . Test your code with comma-delimiter-character and with space delimiter character and see the difference)

Sample Run:

```
mark1 = 100
mark2 = 97
mark3 = 85
The average is 94
```

Solution:



```
//Lab9-Task1 - by: Mona Altassan
#include <fstream>
#include <iostream>
using namespace std;
int main(){
    float sum=0, avg, mark;
    int count=0;

    //Open file streams for reading & writing
    ifstream fin("Scores.txt");
    ofstream fout("Average.txt");

    //check if files were opened successfully
    if (!fin.is_open() || !fout.is_open()){
        cout<<"\nError: Unable to open file(s)!"<<endl;
        return -1;
    }

    //Reads the marks from the input file
    while (!fin.eof()){
        fin>>mark;
        fin.ignore(1);
        sum+=mark;
        count++;
        cout << "\nmark"<<count<<" = " <<mark;
    }

    avg=sum/count;

    //Write the average to the output file
    cout<<"\nThe average for "<<count<<" marks is "<<avg;
    fout<<"\nThe average for "<<count<<" marks is "<<avg;

    // Close the files
    fin.close();    fout.close();
    return 0;
}
```



Task#2:

Write a C++ program that saves records to a sequential access file named (numbers.txt). Each record should contain two fields separated by the '#' character. The first field should contain numbers from 10 through 25. The second field should contain the square of the number in the first field. For example, the first record will contain the number 10, the '#' character, and the number 100. After running your program, the Numbers.txt file should contain the following:

Numberx.txt

```
10#100
11#121
12#144
13#169
14#196
15#225
16#256
17#289
18#324
19#361
20#400
21#441
22#484
23#529
24#576
25#625
```

Solution:

```
//lab9-Task2
#include <cmath>
#include <iostream>
#include <fstream>
using namespace std;

int main()
{
    ofstream write;
    write.open("numbers.txt", ios::out);

    if (write.is_open())
    {
        for (int x = 10; x < 26; x++)
            write << x << '#' << pow(x, 2) << endl;

        write.close();
    }else
        cout << "The file could not be opened!" << endl;
    return 0;
} //end of main
```



Task#3:

Write a program that allows the user to record the names of cities and their corresponding ZIP codes in a sequential access file named (cities.txt). The program also should allow the user to look up a ZIP code in the file and display the name of its corresponding city. If the ZIP code is not in the file, the program should display an appropriate message.

ZIP code	City
60561	Darien
60544	Hinsdale
60137	Glen Ellyn
60135	Downers Grove
60136	Burr Ridge

Solution

```
//Lab9- Task3 - by: Mona Altassan

#include <iostream>
#include <fstream>
using namespace std;

void addCity()
{
    string name;
    int zipcode, choice;
    ofstream write("cities.txt", ios::app);

    if (!write.is_open())
    {
        cout<<"\nError: Unable to open the file!";
        return;
    } //end if

    do{
        cout<<"\nEnter the zip code for the city: ";
        cin>>zipcode;
        write<<zipcode<<" ";
        cin.ignore(); //ignore the new line entered since there is a
        getline after this

        cout<<"Enter the name of the city: ";
        getline(cin, name);
        write<<name<<endl;

        cout<<"Would you like to add another city? (-1 to quit) ";
        cin>>choice;
    }while(choice!=-1);

    write.close();
} //end of addCity
```



```
void searchCity()
{
    ifstream read("cities.txt");
    int zipcode, zip;
    string city;

    if (!read.is_open())
    {
        cout<<"\nError: Unable to open the file!";
        return;
    } //end if

    cout<<"\nEnter the zip code you would like to search for: ";
    cin>>zip;

    while(!read.eof())
    {
        read>>zipcode;
        getline(read, city);

        if (zipcode==zip)
        {
            cout<<"The city is "<<city<<endl;
            read.close();
            return;
        } //end if
    } //end while

    cout<<"Not found"<<endl;
    read.close();
} //end of searchCity

int main()
{
    int choice;
    do{
        cout<<"\n----- Menu -----"
            <<"\n1- Add cities to the file\n2- Search for a city\n3- Exit"
            <<"\n-----\n";
        cin>>choice;

        if(choice ==1)
            addCity();
        else if(choice ==2)
            searchCity();

    }while(choice!=3);
    return 0;
} //end of main
```