Programming Fundamental

Lab Manual - Week 12

Introduction

Welcome Back to your favorite Programming Lab students. In this lab manual, we shall work together to learn and implement new programming concepts.

Skill: learning fstream to Read, Write and Append in files

Let's do some coding.

Introduction

We have learned how to create programs that store user-entered data and can use this data to carry out different tasks. However, a key limitation was that all the information stored in various variables was lost once the program was closed. Therefore, we were unable to store the data permanently on the computer. This week, we will discover how to store data permanently on the computer.

Task 01(WP): Write a program that writes "this is sample text" to a file.

Step 01: Include the fstream library in your program

```
# include <fstream>
```

Step 02: Create a file variable in your program

```
fstream file;
```

Step 03: Open the file in desired mode using the variable created in the previous step

```
file.open("example.txt", ios::out);
```

We can open the file in one of the following modes

File Mode	Description
ios::in	We use the mode when we want to read from the file
ios::out	We use the mode when we want to write to the file
ios::app	We use the mode when we want to append data in the file

Step 04: Write to the text using a variable or a string.

file << "This is sample text";
OR
string text = "This is sample text";
file << text;

Step 05: Close the file when you have performed your desired operations on the file.

file.close();

Task 02(OP):

(a) Write a program that takes an integer from the user and writes to a file.
(b) Write a program that takes a decimal input from the user and writes to a file.
(c) Write a program that takes a character input from the user and writes to a file.

Hint: Remember the five steps to use a file.

Task 03(CL): Write a program that reads the name stored in the file and prints in on screen.

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    string name;
    fstream file;
    file.open("example.txt", ios::in);
    file >> name;
    file.close();
    cout << "The name in file: " << name;
}
```

In this solution, we have **opened the file in the read mode** and stored the string from the file in the **name** variable.

Task 04(OP):

(a) Write a program that reads an integer stored in the file and prints in on screen.
(b) Write a program that reads a decimal stored in the file and prints in on screen.
(a) Write a program that reads a character stored in the file and prints in on screen.

Hint: Remember the five steps to use a file.

Question:

Now, what if we want to read a complete line (with the spaces) from the file ? Can we use the same technique for reading from the file as we have learned earlier in the manual ?

Consider the following task for better understanding.

Task 05(WP): Write a program that reads a complete line stored in the file.

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    string line;
    fstream file;
    file.open("example.txt", ios::in);
    getline(file,line);
    file.close();
    cout << "The Line in file: " << endl << line;
}
```

In this solution, we have used the getline(file,line) function to take input from the file that will return a complete line in the our variable.

Programming Fundamental

Lab Manual - Week 12

Consider that we want to read a file with multiple lines and we want to display all its contents on the console screen.

Task 06(WP): Write a c++ program that displays all the contents of a file.

```cpp
#include <iostream>
#include <fstream>
using namespace std;
main()
{
    string line;
    fstream file;
    file.open("example.txt", ios::in);
    while(!file.eof())
    {
        getline(file,line);
        cout << line<<endl;
    }
    file.close();
}
```

Task 07(OP): Write a program that read the text file line by line and display total number of lines on user screen.

Task 08(OP): Write a program that read text file character by character and display total number of characters in the file.

Task 09(OP): Write a program that calculates the frequency of character in file.

Task 10(CP): You are assigned to develop a project in which the project manager wants the following functionalities.

- Create a file name **student.txt** in your Directory using the C++.
- Create a function that will ask the student's details from the console
- Create a separate function to save that information in the separate line of the file student.txt.

  Information contains:

  1. Student Name
  2. Student Age
  3. Student Matric marks
  4. Student Fsc marks

Skill: learning fstream to Read, Write and Append in files

5. Student Ecat Marks

Task 11(CP):

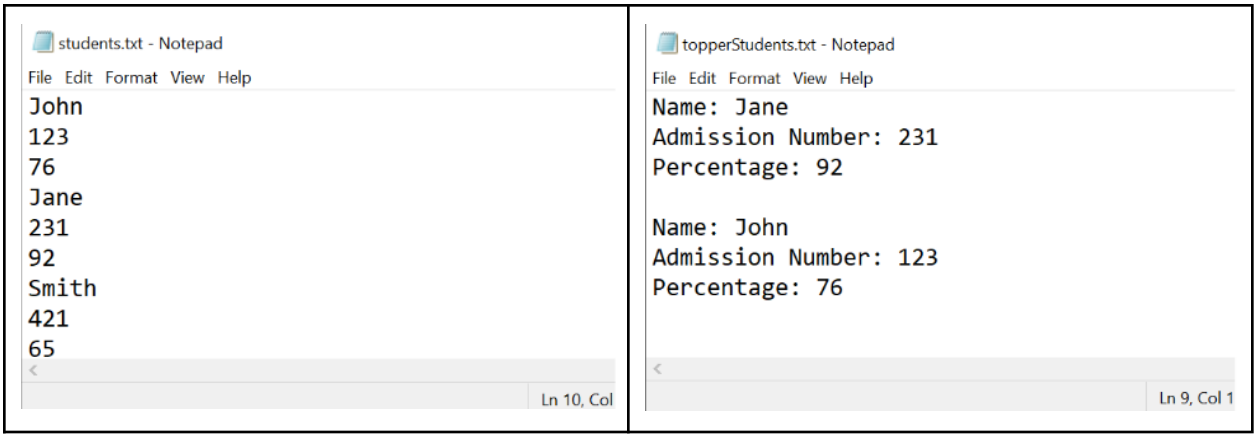
Develop a Signup Application using File System.

- As a user, when I SignUp to the system the username and password stores into the file.
- As a user, when I SignIn to the system the program should let me in if i am a valid user.

Task 12(CP):

Write a program that reads the data from the students.txt file and stores the data of students in the topperStudents.txt.

Instruction: Store the data of only those students having percentage higger than 60.



Task 13(CP): Write a function in C++ to count the number of lines from a text file "story.txt" which is not starting with an alphabet "T".

Example: If the file "story.txt" contains the following lines:
A boy is playing there. There is a playground. An aeroplane is in the sky. The sky is pink. Alphabets and numbers are allowed in the password.

Task 14(CP):

Write a function display_words() in c++ to read lines from a text file "story.txt", and display those words, which are less than 4 characters.

Skill: learning fstream to Read, Write and Append in files

Programming Fundamental

Lab Manual - Week 12

Good Luck and Best Wishes !!

Happy Coding ahead :)

Skill: learning fstream to Read, Write and Append in files

Muhammad Irzam & Maida Shahid, Department of Computer Science, UET Lahore