Name: _______Hong Hai Dang Nguyen_______ Student ID:____103503191 ____

SWE20004 Technical Software Development

Lab 7 (week 7)

You will need:
Online C++ IDE or Computer installed IDE
A computer with internet access

In this lab you will investigate C++ File I/O

Before you start the lab exercise, download an appropriate C++ IDE to run your commands and programs.

1. Answer the following questions? Don't copy and paste – write down what it is - in your own words.

   A. Which class can be used to create files that can be written to, but not read from?

The class that is used to create files that can be written to, but not read from, is the ofstream. This data type, which is used to create files and write data to them, represents the output file stream.

   B. Which class can be used to create files that can be both written to and read from?

The class that can be used to create files that can be both written to and read from is the fstream. This data type represents the file stream in general and possesses both ofstream and ifstream's functionality, allowing it to create files, write data to them, and read data from them.

   C. Which method reads in an entire line from a text file?

getline(file_name, line);
ifstream file_name;
string line;

   D. Which method writes a single character to a text file?

ofstream textFile; textFile<< ‘x’; //x is the single character that is written to textFile

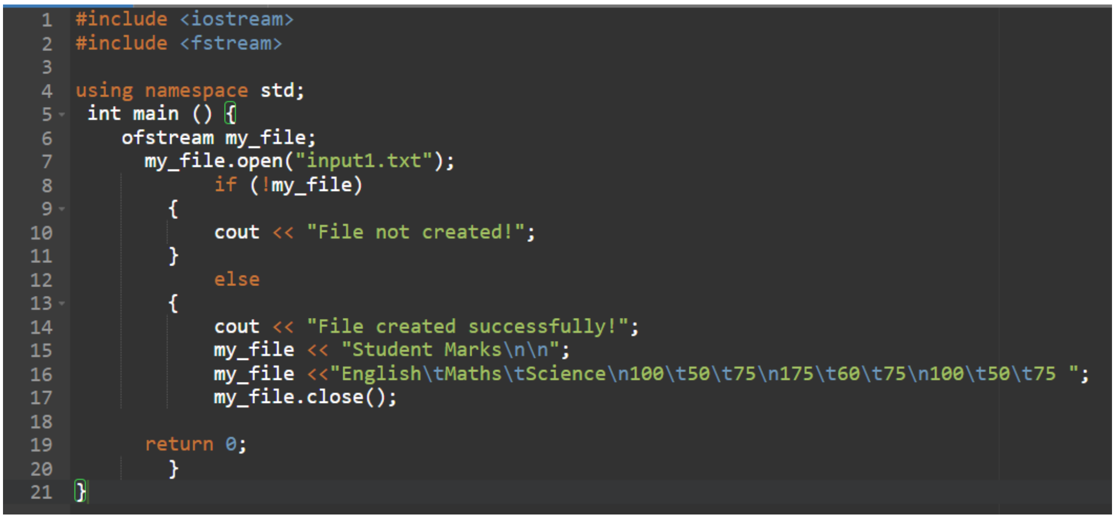
2. What is displayed by the following code? Please debug errors in the code below and fix it.

   The error is highlighted in red next to the code

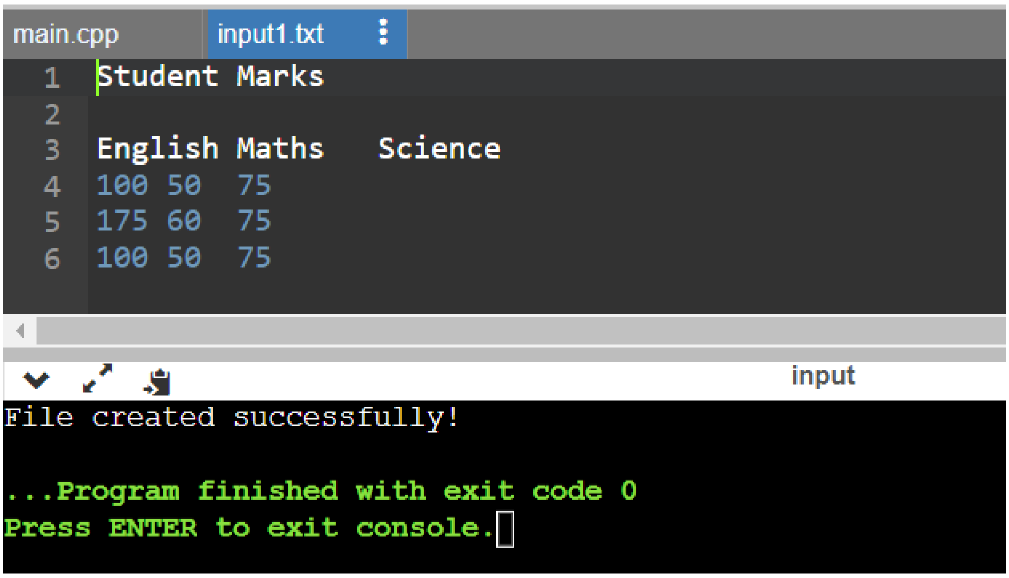
□ #include <iostream>

□ #include <iostream> **(Repitition, it should be #include <fstream>)**

```
□
□    using namespace std;
□
□    int main () {
□      ofstream my_file;
□      my_file.open("input1.txt");
□                  if (!my_file)
□                  {
□                              cout << "File not created!";
□                  }
□                  else
□                  {
□                              cout << "File created successfully!";
□                              my_file << "Student Marks\n\n";
□                              my_file <<"English\tMaths\tScience\n100\t50\t75\n175\t60\t75\n100\t50\t75 ";
□
□                              my_file.close();
□
□      return 0;
□    }
□    (Missing a curly bracket at the end)
```



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

using namespace std;
 int main () {
    ofstream my_file;
      my_file.open("input1.txt");
            if (!my_file)
        {
            cout << "File not created!";
        }
            else
        {
            cout << "File created successfully!";
            my_file << "Student Marks\n\n";
            my_file <<"English\tMaths\tScience\n100\t50\t75\n175\t60\t75\n100\t50\t75 ";
            my_file.close();

      return 0;
        }
}
```



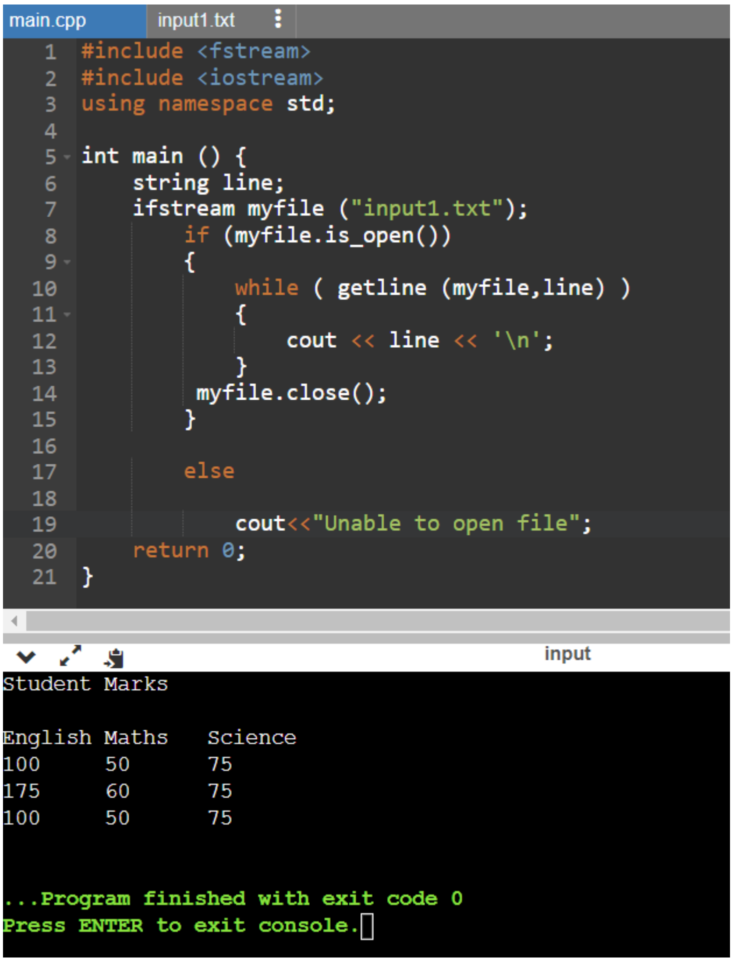
```
Student Marks

English Maths   Science
100 50  75
175 60  75
100 50  75
```

```
File created successfully!

...Program finished with exit code 0
Press ENTER to exit console.
```

3. What is displayed by the following code? Please debug errors in the code below and fix it.

   The error is highlighted in red next to the code. This is the continue code to question 2, as the input1.txt was created.

```
□  #include <fstream>
□  (Missing the iostream library)
□  using namespace std;
□
□  int main () {
□    string line;
□    ifstream myfile ("input1.txt");
□    if (myfile.is_open()) (Missing an open curly bracket)
□      while ( getline (myfile,line) )
□      {
□        cout << ‘line’ << '\n'; (No quotation mark is needed)
□      }
□      myfile.close();
□    (Missing a closing curly bracket)
□    else cout << "Unable to open file";
□    return 0;
□    (Missing an ending curly bracket)
```



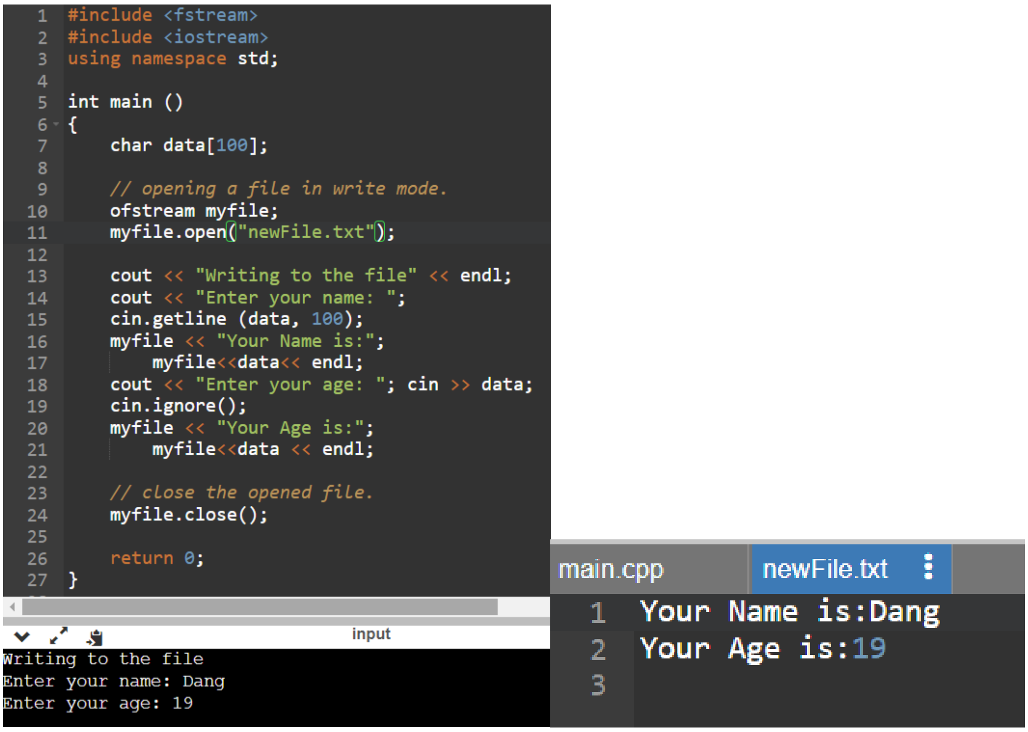
```
#include <fstream>
#include <iostream>
using namespace std;

int main () {
    string line;
    ifstream myfile ("input1.txt");
        if (myfile.is_open())
        {
            while ( getline (myfile,line) )
            {
                cout << line << '\n';
            }
          myfile.close();
        }

        else

            cout<<"Unable to open file";
    return 0;
}
```

```
Student Marks

English Maths   Science
100     50      75
175     60      75
100     50      75


...Program finished with exit code 0
Press ENTER to exit console.
```

4. What is displayed by the following code? Please debug errors in the code below and fix it.

   The error is highlighted in red next to the code

```
□  #include <fstream>
□  #include <iostream>
□  using namespace std;
□
□  int main ()
□  {
□     char data[100];
□
□     // opening a file in write mode.
□     ofstream myfile;
□     myfile.open "newFile.txt"; (Missing round brackets for: "newFIle.txt")
□
□     cout << "Writing to the file" << endl;
□     cout << "Enter your name: ";
□     cin.getline data, 100; (Missing round brackets for: data,100)
□     myfile << "Your Name is:";
□               myfile<<'data' << endl; (No quotation mark is needed)
□     cout << "Enter your age: "; cin >> data;
□     cin.ignore();
□     myfile << "Your Age is:";
□               myfile<<data << endl;
□
□     // close the opened file.
□     myfile.close();
□
□    return 0;
□  }*/ (Only the ending curly bracket is needed)
```



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

int main ()
{
    char data[100];

    // opening a file in write mode.
    ofstream myfile;
    myfile.open("newFile.txt");

    cout << "Writing to the file" << endl;
    cout << "Enter your name: ";
    cin.getline (data, 100);
    myfile << "Your Name is:";
        myfile<<data<< endl;
    cout << "Enter your age: "; cin >> data;
    cin.ignore();
    myfile << "Your Age is:";
        myfile<<data << endl;

    // close the opened file.
    myfile.close();

    return 0;
}
```

```
Writing to the file
Enter your name: Dang
Enter your age: 19
```

```
Your Name is:Dang
Your Age is:19

```

5. Create a file I/O for a database based on the following conditions

    - Create student record
        - Create values such as name, student ID, Course details, units and grades.
        - Save the data in student.txt file (don't overwrite)
    - Create facilitator/tutor record
        - Create values such as name, staff ID, Course details, units and class timetable.
        - Save the data in staff.txt file (don't overwrite)

    When you run your code, you need display/enter new data in to the database based on the entry options below

    Press number 1 – Enter a new student details to the record

    Press number 2 - Enter a new staff details to the record

    Press number 3 – Display student record

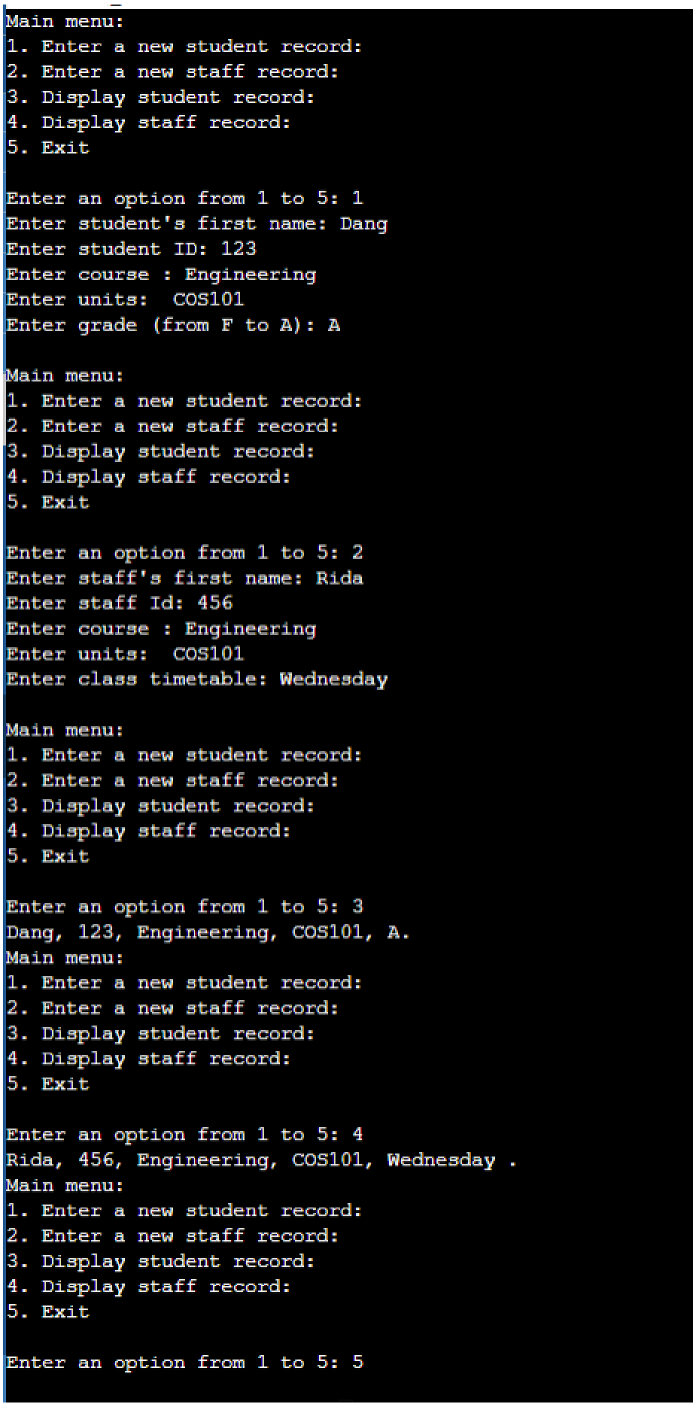
    Press number 4 - Display staff record

    Note: we need to see those options as part of your code.

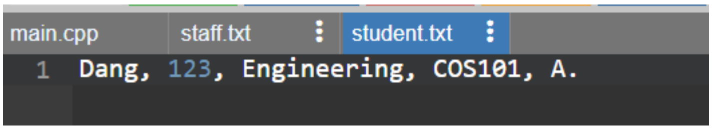
```
#include<iostream>
#include<fstream>
#include<string>

using namespace std;

int main(){
    fstream student, staff;
    // Open File
    student.open("student.txt",ios::in | ios::out | ios::app);
    staff.open("staff.txt",ios::in | ios::out | ios::app);
    if(!student || !staff)
    {
        cout<<"Error in creating file. Please try again";
        return 0;
    }
    // Menu
    while(true){
        int option;
        cout<<"Main menu:"<<endl;
        cout<<"1. Enter a new student record: "<<endl;
        cout<<"2. Enter a new staff record: "<<endl;
        cout<<"3. Display student record: "<<endl;
        cout<<"4. Display staff record: "<<endl;
```

```
cout<<"5. Exit "<<endl;
cout<<" "<<endl;
cout<<"Enter an option from 1 to 5: ";
cin>>option;
if(option == 1){
    string result,input;
    result = "";
    cout<<"Enter student's first name: ";cin>>input;
    result += input;
    cout<<"Enter student ID: ";cin>>input;
    result += ", "+input;
    cout<<"Enter course : ";cin>>input;
    result += ", "+input;
    cout<<"Enter units:  ";cin>>input;
    result += ", "+input;
    cout<<"Enter grade (from F to A): ";cin>>input;
    cout<<"\n";
    result += ", "+input;

    student.seekp(0, std::ios_base::end);
    student.clear();
    student << result+". ";
}else if(option == 2){
    string result,input;
    result = "";
    cout<<"Enter staff's first name: ";cin>>input;
    result+=input;
    cout<<"Enter staff Id: ";cin>>input;
    result += ", "+input;
    cout<<"Enter course : ";cin>>input;
    result += ", "+input;
    cout<<"Enter units:  ";cin>>input;
    result += ", "+input;
    cout<<"Enter class timetable: ";cin>>input;
    cout<<"\n";
    result += ", "+input+" ";

    staff.seekp(0, std::ios_base::end);
    staff.clear();
    staff << result+". ";
}else if(option == 3){
    int cnt = 0;

    student.seekg(0,  std::ios_base::beg);
    for(string line; getline(student, line) ; ) {
        cout << line << endl;
        cnt++;
    }
```

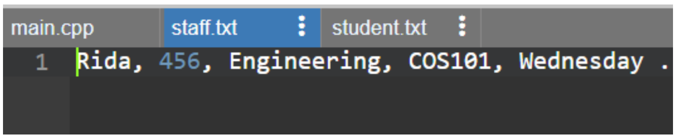
```
            if(cnt == 0)
            cout<<"There is no record."<<"";

        }else if(option ==4 ){
            int cnt = 0;

            staff.seekg(0,  std::ios_base::beg);
            for(string line; getline(staff, line) ; ) {
                cout << line << endl;
                cnt++;
            }
            if(cnt == 0)
            cout<<"There is no record."<<"";
        }else if(option == 5){
            break;
        }else{
            cout<<"Please select again."<<endl;
        }
    }
    student.close();
    staff.close();
}
```



```
Main menu:
1. Enter a new student record:
2. Enter a new staff record:
3. Display student record:
4. Display staff record:
5. Exit

Enter an option from 1 to 5: 1
Enter student's first name: Dang
Enter student ID: 123
Enter course : Engineering
Enter units:  COS101
Enter grade (from F to A): A

Main menu:
1. Enter a new student record:
2. Enter a new staff record:
3. Display student record:
4. Display staff record:
5. Exit

Enter an option from 1 to 5: 2
Enter staff's first name: Rida
Enter staff Id: 456
Enter course : Engineering
Enter units:  COS101
Enter class timetable: Wednesday

Main menu:
1. Enter a new student record:
2. Enter a new staff record:
3. Display student record:
4. Display staff record:
5. Exit

Enter an option from 1 to 5: 3
Dang, 123, Engineering, COS101, A.
Main menu:
1. Enter a new student record:
2. Enter a new staff record:
3. Display student record:
4. Display staff record:
5. Exit

Enter an option from 1 to 5: 4
Rida, 456, Engineering, COS101, Wednesday .
Main menu:
1. Enter a new student record:
2. Enter a new staff record:
3. Display student record:
4. Display staff record:
5. Exit

Enter an option from 1 to 5: 5
```



```
Dang, 123, Engineering, COS101, A.
```



```
Rida, 456, Engineering, COS101, Wednesday .
```

Name: _______Hong Hai Dang Nguyen_______ Student ID:____103503191 ____

Report (SWE20004)

Write a one-page report on this lab covering the following:

1. Summarize the topics you explored and the activities you did during this lab.
2. Classify (group) these topics and actions under appropriate headings. Do not just copy the headings used in the instructions. For example, explain the following, what are the following commands do?
3. Discuss the relevance of these topics and actions in terms of C++ programming. i.e. How do the things in this lab work contribute to your understanding of C++ programming overall?
4. Why do you need to understand (and use) C++ **File I/O**?

This report is worth 5% towards your unit assessment. Use the below page as a template. Either you can type it or write it in your words.

Introduction

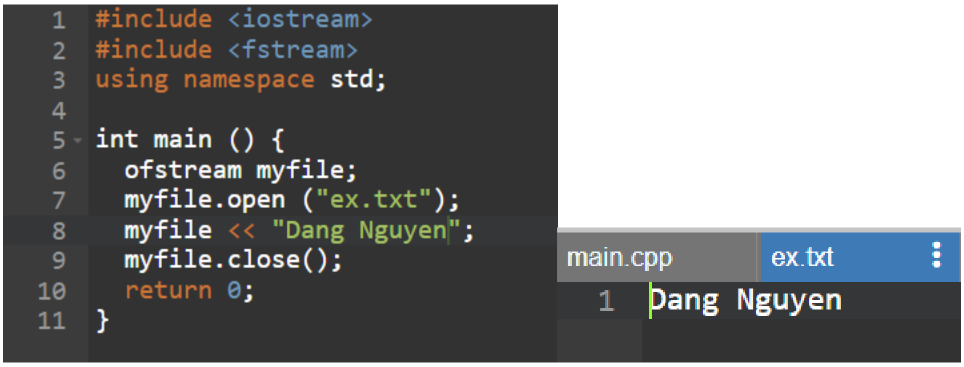
This week’s report will illustrate the topics covered and the activities of the lab. More specifically, we were taught about the Input/output with files (or File I/O), how to open and close a file. In addition, we went over some of the exercises in this week's lab in order to strengthen our knowledge and understanding of the topics.

Input/output with files

There are 3 classes available in C++ for character output and input to/from files:

- ofstream: A file-writing stream class
- ifstream: A class for reading files from streams.
- fstream: A stream class that can read and write to/from files.

From the classes ***istream*** and ***ostream***, these classes are either directly or indirectly descended. ***cin*** is an object of class istream, and ***cout*** is an object of class ostream, both of which we have already learnt and used. Since our file streams are related, we have already started using classes that are related to them. In reality, we can use our file streams in the same way that we use ***cout*** and ***cin***, with the exception that we must link these streams to actual physical files. An example program for this is:



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

int main () {
  ofstream myfile;
  myfile.open ("ex.txt");
  myfile << "Dang Nguyen";
  myfile.close();
  return 0;
}
```

```
Dang Nguyen
```

Open a file

Associating an object from one of these classes with a real file is typically the first action taken on it. The term "open a file" refers to this action. In a program, an open file is represented by a stream, which is an object of one of these classes (in the previous example, this was myfile). Any input or output operations carried out on this stream object will be applied to the corresponding physical file.

To open a file with a stream object, we use the member function ***open***:

open (filename, mode);

Where ***filename*** is a string identifying the name of the file to be opened, and ***mode*** is a parameter that may be optionally specified with a fusion of the following flags:

`ios::in`	Open for input operations.
`ios::out`	Open for output operations.
`ios::binary`	Open in binary mode.
`ios::ate`	Place the file's starting point at the end of the file. If this option is not set, the file's first position is used as the initial position.
`ios::app`	All output operations are carried out at the file's end, adding the content to the existing content.
`ios::trunc`	If the file is opened for output operations and it previously has content, the old material is removed and the new content is added.

If the file is opened without a second parameter, the default mode of each of the open member functions of classes ofstream, ifstream, and fstream is used:

Class	**Default mode parameter**
`ofstream`	**ios::out**
`ifstream`	**ios::in**
`fstream`	**ios::in | ios::out**

ios::in and ***ios::out*** are automatically and respectively assumed for the ***ifstream*** and ***ofstream*** classes, even if a mode that does not include them is supplied as the second parameter to the open member function (the flags are combined).

Only when the function is called without passing any values for the mode parameter will the default value for ***fstream*** be used. The default mode is overridden rather than combined if the function is invoked with any value for that parameter.

Close a file

A file must be closed after input and output operations are complete in order to alert the operating system and make its resources available once again. To do it, we use the member function ***close*** of the stream. This member function opens the file and flushes any associated buffers. The file is now again available to be opened by other processes after this member function has been called, and the stream object can then be used to open another file. If an object is destroyed while it is still linked to an open file, the member function ***close*** is automatically called by the destructor.

Lab activities

The first 4 exercises of this week’s lab exercises were rather conventional and can be completed easily, yet the final question of this lab is significantly challenging, as there are many lines of code and every mistake, even the smallest one, can make the program unexecutable. The final question took me a long time to finish it, which I argue that it is the most difficult and time-consuming question so far in this unit.

The importance of this week’s topic

File I/O can help user to store their data into files and to keep updating the files with new contents. Therefore, having a deep understanding of File I/O is essential for every programming language (including C++), as the programmers have to manage a huge number of different files and they must be efficient in handling files to be able to finish their tasks more quickly.

Conclusion

Overall, this week's lab helped us to understand the Input/output with files and how to open and close a file. This, from my perspective, is the hardest aspect so far in this unit. I would say that I’m currently not really confident in implementing this into my program, but given that I have understood the fundamentals, I shall be able to provide inputs and receive outputs from files easier in the future.