CS1002 – Programming Fundamentals

Lecture # 27
Monday, December 05, 2022
FALL 2022
FAST – NUCES, Faisalabad Campus

Muhammad Yousaf

What is a File?

- A computer file
  - Collection of bytes
  - Hold information
  - Stored permanently on a secondary storage device (e.g., disk)
- Types of files
  - **Text File:** A stream of characters to process sequentially by a computer
  - **Image:** A visual presentation of any entity
  - **Media:** Audio/Video file
  - **Binary:** A non-text file. Mostly refers to a file that can be interpreted by a software or hardware

Filing in programming

- **Computer Program**
  - A process of step by step instructions to perform specified task and to produce result on given input
  - File can be used to provide input data to a program or receive output data from a program, or both
  - Reading a file from secondary storage
  - Writing a file permanently for future

Why File Handling in programming?

- Convenient way to deal large quantities of data
- Store data permanently (until file is deleted)
- Avoid typing data into program multiple times
- Share data between programs
- Printable reports
- Programming languages provide significant support for file processing
- For file handling, we need to know:
    - How to "connect" file to program
    - How to tell the program to read data
    - How to tell the program to write data
    - Error checking and handling **EOF**

Cont’d

- Limitations of Console Input and output
- Input from Keyboard
  - Large data Input
  - Typos mistakes
  - Time consuming & inefficient
- Screen Output
  - Limited view on screen

File Handling in C++

- C++ supports file handling in an attractive way
- Streams are used to communicate with file
  - **Stream of bytes** to do input and output to different devices
- A program can read data from file or write data to file
- File ends with ***End-Of-File (EOF)*** marker
- Five steps for file handling in C++ Language

  I. Include **fstream** header file

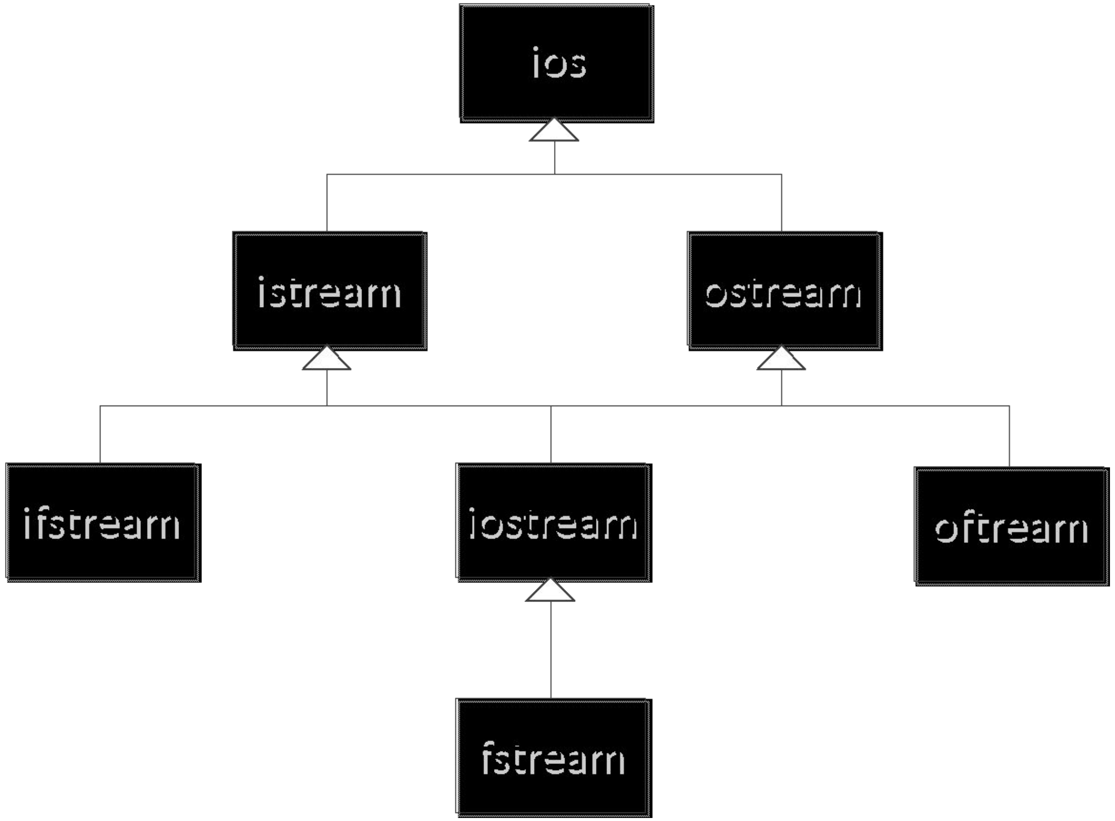
  II. Declare file stream variable(s)

  III. Associate the file stream variable(s) with the input/output source(s)

  IV. Performs Read/Write operations

  V. Close the file(s)

Streams Hierarchy in C++



- **ios** is the base & abstract class
- **istream** and **ostream** inherit **ios**
- **ifstream** inherits **istream**
- **ofstream** inherits **ostream**
- **iostream** inherits **istream** and **ostream**
- **fstream** inherits **iostream**

C++ File Stream Functions

Function	Description
`open()`	To open a file to read or write
`is_open()`	To test file either open or not
`eof()`	To check in reading a file either marker reach End-Of-File (EOF)
`close()`	To close the file
`>>`	Read data from file in general (operator)
`<<`	Write data in file in general (operator)
`getline()`	Reading a single line

Program Skelton for File Processing

```cpp
#include <fstream> // the header file/class for file stream objects
using namespace std;
int main()
{
    //Declare file stream variables such as the following
    ifstream my_input_file;  //An input file stream object
    ofstream my_output_file; //An output file stream object

    //Open the files
    my_input_file.open("prog.dat"); //open the input file
    my_output_file.open("prog.out"); //open the output file

    //Code for data manipulation

    //Close files
    my_input_file.close();  //Close the file associated with this stream
    my_output_file.close(); //Close the file associated with this stream
    return 0;
}
```

Reading from a File

```
#include<iostream>
#include<fstream>
using namespace std;
int main(){
    ifstream my_input_file;
    my_input_file.open("myData.txt");
    if(!(my_input_file.is_open()))
    {
        cout<<"File cannot be opened.";
        return 0;
    }
    cout<<"File Contents: \n";
    char ch;
    while(!my_input_file.eof())
    {
        my_input_file.get(ch); // using get() function
        cout << ch;
    }
    my_input_file.close();
    return 0;

}
```

```
Input File: myData.txt
Reading a text file. Thank You.
```

Last character will be printed twice???

```
Output
File Contents:
Reading a text file. Thank You..
```

Reading from a File

```
#include<iostream>
#include<fstream>
using namespace std;
int main()
    {ifstream my_input_file;
    my_input_file.open("myData.txt");
    if(!(my_input_file.is_open()))
    {
        cout<<"File cannot be opened.";
        return 0;
    }
    cout<<"File Contents: \n";
    char ch;
    while(!my_input_file.eof())
    {
        my_input_file.get(ch); // using get() function
        if(!my_input_file.eof())
            cout << ch;
    }
    my_input_file.close();
    return 0;

}
```

```
Input File: myData.txt
Reading a text file. Thank You.
```

Solution

```
Output
File Contents:
Reading a text file. Thank You.
```

Writing to a File

```
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
    ofstream my_output_file;
    my_output_file.open("myData.txt");
    if(!(my_output_file.is_open()))
    {
        cout << "File cannot open.";
        return 0;
    }
    cout << "Writing contents to file: \n";
    do
    {
        ch = getchar();
        my_output_file << ch;
    } while(ch!='.');
    my_output_file.close();
    return 0;
}
```

Purpose:
This program take input from user and full stop (.) to end. Then write the entered data in a text file.

Sample Output

```
Writing contents to file:
Trying to write in test file.
```

More I/O Functions & Sample Programs

Sample Problem - I

- Write a program, which reads an input file of employee's i.e. "employeein.txt", add 200 to the salary of each employee, and write the result in a new file "employeeout.txt".

The sample input file "employeein.txt"	*The output file "employeeout.txt"*
Aamir 12000	**Name Salary**
Amara 15000	Aamir 14000
Adnan 13000	Amara 17000
Afzal 11500	Adnan 15000
	Afzal 13500

Analysis & Design

- Input
    - Employee Names and Salaries
- Output
    - Employee Name & Updated Salary
- Design of Algorithm
    - Define input & output stream variables
    - Open input (employeein.txt) & output (employeeout.txt) files
    - Get data from input file (Name , Salary) of each employee
    - Update salary by adding 2000 to original salary
    - Write Name and Updated Salary to output file of each employee
    - Close the files
    - Test Your program for different input files of same structure

Solution

```
#include<iostream>
#include<fstream>
#include<string>
#include<conio.h>
using namespace std;
int main()
{
    ifstream inData;
    ofstream outData;
    string name;
    int salary;
    inData.open("employeein.txt");
    outData.open("employeeout.txt");
    if(!inData)
    {
        cout << "Can't open input file" <<
endl;
        return 0;
    }
    if(!outData)
    {
        cout << "Can't open Output file“ <<
endl;
    }
    outData << "Name" << "\t" << "Salary" << endl;
    while(inData) // while(!inData.eof())
    {
        inData >> name;
        inData >> salary;
        outData << name << "\t"
                << salary+2000 << endl;
    }
    inData.close();
    outData.close();
    system("Pause");
    system("employeeout.txt");
    return 0;
}
```

```
Input File: employeein.txt
Aamir 12000
Amara 15000
Adnan 13000
Afzal 11500
```

```
Output File: employeeout.txt
NameSalary
Aamir   14000
Amara   17000
Adnan   15000
Afzal   12500
```

Sample Problem - II

```
//single input string and display
#include<iostream>
#include<fstream>
#include<string>
using namespace std;
void main()  {
    //declaration
    string Name;
    ofstream out;
    ifstream in;
    //inserting single string with //space
    out.open("test.txt");
    getline(cin, Name);
    out << Name << endl;
    out.close();
    // displaying string
    in.open("test.txt");
    getline(in,Name);
    cout << Name << endl;
    in.close();
    system("pause");
}
```

Sample Problem - III

```
//Append Mode, Get all data from file
#include<iostream>
#include<fstream>
#include<string>

using namespace std;
void main()
{
    ofstream out;
    ifstream in;
    string name;
    out.open("test.txt", ios::app);
    for (int i = 0 ; i < 10 ; ++i){
        getline(cin,name);
        out << name << endl;
    }
    out.close();
    in.open("test.txt");
    while (!in.eof()) {
        getline(in, name);
        cout << name << endl;
    }
    in.close();
}
```

Reading Material

- C++ Programming: From Problem Analysis to Program Design, 5th Edition by D.S. Malik
- C++ How to Program, 8th Edition by Deitel & Deitel
- Cplusplus [Online] http://www.cplusplus.com/
- http://www.tutorialspoint.com/cplusplus/cpp_files_streams.htm
- http://www.cppforschool.com/tutorial/Files1.html
- http://www.wellho.net/resources/ex.php4?item=c235/file01.cpp
- http://msdn.microsoft.com/en-us/library/d3ccyysc.aspx
- http://www.elearningbio.com/forum/images/RW_20131024105519PM_12_computer_science_notes_CH03_data_file_handling.pdf

Questions