Aug 27, 2015

COMPUTER PROGRAMMING

LAB 3

ABDUL AZIZ

FAST NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

Lab Instructor	**Mr. Abdul Aziz**
Course	**Computer Programming Lab**
Duration	**2hrs**

Objectives:

In this lab, following topics will be covered

- C++ File I/O
- Opening Files
- Reading from/Writing to/ a file
- File Modes
- Closing files and flushing output buffer
- Checking file open success
- End of File
-

So far, we have been using the **iostream** standard library, which provides **cin** and **cout** methods for reading from standard input and writing to standard output respectively. Now we have to use another C++ standard library called **fstream** which defines following new data types.

Data Type	Description
Ofstream	This data type represents the output file stream and is used to create files and to write information to files.
Ifstream	This data type represents the input file stream and is used to read information from files.
Fstream	This data type represents the file stream generally, and has the capabilities of both ofstream and ifstream which means it can create files, write information to files, and read information from files.

1. Opening Files

A file must be opened before you can read from it or write it. Either the **ofstream** or **fstream** object may be used to open a file for writing or **ifstream** object is used to open a file for reading purpose.

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

int main()
{
      ifstream fin; // fin named object created
      fin.open("Input.txt");//
      ofstream fout;
      fout.open(“Output.txt”);
}
```

2. Reading from a File

To read information from a file into your program using the stream extraction operator (>>) just as you use that operator to input information from the keyboard. The only difference is that you use an **ifstream** or **fstream** object instead of the **cin** object.

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

int main()
{
      Char ch;
      ifstream fin;
      fin.open("Input.txt");
      fin>>ch;
      cout<<ch;
}
```

3. Writing to a File

To write information to a file from your program using the stream insertion operator (<<) just as you use that operator to output information to the screen. The only difference is that you use an **ofstream** or **fstream** object instead of the **cout** object.

A simple program for writing to a file and then reading from that file.

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

int main ()
{

  char data[100];

                              // open a file in write mode.
  ofstream outfile;
  outfile.open("afile.txt");

  cout << "Writing to the file" << endl;
  cout << "Enter your name: ";
  cin>> data;

                               // write inputted data into the file.
  outfile << data << endl;

  cout << "Enter your age: ";
  cin >> data;
  cin.ignore();

                               // again write inputted data into the file.
  outfile << data << endl;

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

                              // open a file in read mode.
  ifstream infile;
  infile.open("afile.txt");

  cout << "Reading from the file" << endl;
  infile >> data;

                                // write the data at the screen.
  cout << data << endl;

                                // again read the data from the file and display it.
  infile >> data;
  cout << data << endl;

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

  return 0;
}
```

4. File Modes

4.1 Input Mode:

```
fin.open("Input.txt");
```

is equal to

```
fin.open(“Input.txt” , ios::in);
```

4.2 Output Mode:

```
fout.open("Output.txt");
```

is equal to

```
fout.open(“Output.txt” , ios::out);
```

4.3 Append Mode:

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

int main()
{
      Char ch;
      ifstream fin;
      fin.open("Input.txt");
      ofstream fout;
      fout.open("Output.txt", ios::app);
      fout<<" and ";
      fin>>ch;
      fout<<ch;
}
```

4.4 Truncate Mode:

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

int main()
{
      Char ch;
      ifstream fin;
      fin.open("Input.txt");
      ofstream fout;
      fout.open("Output.txt", ios::trunc);
      fout<<" and ";
```

```
        fin>>ch;
        fout<<ch;
}
```

4.5 Alternate syntax for opening a file

```
ifstream fin;
fin.open("Input.txt");
```

is equal to

```
ifstream fin(“Input.txt”);
```

5. Closing Files and Flushing Output buffer

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

int main()
{
      Char ch;
      ifstream fin;
      fin.open("Input.txt");
      ofstream fout;
      fout.open("Output.txt", ios::trunc);
      fout<<" and ";
      fin>>ch;
      fout<<ch;
      fout.flush();
      fin.close();
      fout.close();
}
```

6. Checking File Open Success

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

using namespace std;

int main()
{
      charch;
      ifstream fin;
      fin.open("Input.txt");
      if(fin.fail())
      {
          cout<<"File opening Failed\n";
          exit(1);
      }
      else
      {
          cout<<"File opened Successfully\n";
      }
}
```

Same thing but with “not operator (!)”

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

using namespace std;

int main()
{
   charch;
   ifstream fin;
   fin.open("Input.txt");
   if(!fin)
   {
         cout<<"File opening Failed\n";
         exit(1);
   }
   else
   {
         cout<<"File opened Successfully\n";
   }
}
```

7. Checking End of File

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

int main()
{
    string next;
    ifstream fin("Input.txt");
    fin>>next;
    while(!fin.eof())
    {
        cout<<next<<endl;
```

```
                fin>>next;
        }
}
```

- **Alternate way for checking EOF (End of File)**

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

int main()
{
        string next;
        ifstream fin("Input.txt");
        while(fin>>next)
        {
                cout<<next<<endl;
        }
}
```

Exercise

1. **Write a C++ program to write numbers from 1 to 100 in a data file NOTES.TXT**

2. **Write a C++ program to read the content from a text file OUT.TXT, count and display the number of alphabets present in it.**

3. **Write a C++ program to count the number of blank spaces/white spaces present in a text file named "OUTPUT.TXT".**

4. **Assuming that a text file named FIRST.TXT contains some text written into it, write a program that reads the file FIRST.TXT and creates a new file named SECOND.TXT, to contain only those words from the file FIRST.TXT which start with a lowercase vowel (i.e with 'a', 'e', 'i', 'o', 'u').**
   For example, if the file FIRST.TXT contains
   Carry umbrella and overcoat when it rains
   Then the file SECOND.TXT shall contain
   umbrella and overcoat it

5. **Write a program which reads a number from a file, calculate its factorial, matches it and output on another file whether the factorial is “right” or “wrong”.**

Input	Output
0 0	Wrong
1 1	Right
4 24	Right
5 160	Wrong
7 5040	Right