National Chiao Tung University Laboratory Manual #12
Department of Electrical Computer and Engineering June 9, 2015

UEE 1303(1069): Object-Oriented Programming

Lab #12: File I/O and Exercise

In this laboratory session you will understand how to use file I/O stream.

Lab 12-1: File Input Stream: IFSTREAM

✓ Program `lab12-1` provides an example of using for file input stream to read information from the file `lab12-1.txt`. Here below shows the content of file `lab12-1.txt`.

```
10 20 30 40 50 60 70 80 90 100
```

✓ Program `lab12-1` asks the user to input the filename to open and check if the file can be opened successfully.

```
//File: lab12-1.cpp
#include <fstream>
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string filename;
    cout << "Enter the filename:";
    cin >> filename;
    ifstream in_file;
    in_file.open(filename.c_str());
    if (in_file.fail())
    {
        cout << "File " << filename << " does not exist !!"
             << endl;
        exit(1);
    }
    int num;
    int count = 0;

    while (in_file >> num)
    {
        cout << count << ": " << num << endl;
        ++count;
    }
    in_file.close();
    return 0;
}
```

- Since the member function open takes a c-string variable as its argument, you can use the string member function `c_str()` to convert the data type from string to c-string.
- The statement while (`in_file >> num`) returns `true` if the read operation succeeds and returns `false` when the program attempts to read beyond the end of the file.

Lab 12-2: File Input Stream and `getline`

- ✓ Program `lab12-2` gives an example to read the text file `lab12-2.txt`.
- ✓ Here below shows the content of file `lab12-2.txt`.

```
  When your program takes input from a file, it is said
to be reading from the file; when your program sends output
to a file, it is said to be writing to the file.
```

- ✓ Program `lab12-2` uses a different method to open the file.

```
// file: lab12-2.cpp
#include <fstream>
#include <iostream>
#include <string>
using namespace std;

int main(int argc, char *argv[])
{
    char *filename = argv[1];
    ifstream in_file(filename);
    if (in_file.fail())
    {
        cout << "File " << filename << " does not exist !!"
             << endl;
        exit(1);
    }
    string textline; int count = 0;
    while (getline(in_file,textline,'\n'))
    {
        cout << count << ": " << textline << endl;
        ++count;
    }
    in_file.close(); return 0;
}
```

- Use `./lab12-2 lab12-2.txt` to examine the results of this program
- The statement `getline(in_file,textline,'\n')` returns `true` if the read operation succeeds and returns `false` when the program attempts to read beyond the end of the file.
- Function `good()` indicates that the file is available to be manipulated. Function `fail()` returns `true` if the file does not exist. Function `bad()` indicates that the hardware is bad to read or write the file. Function `eof()` means that the stream reaches the end of file.

Lab 12-3: File Output Stream: OFSTREAM

✓ Program `lab12-3` gives an example to append text in an existing file `lab12-2.txt`.

```
// file: lab12-3.cpp
#include <fstream>
#include <iostream>
using namespace std;

int main()
{
    ofstream myfile;
    myfile.open("lab12-2.txt",ios::app);
    myfile << "new words" << endl;
    myfile.close();
    return 0;
}
```

- `ios::in`, `ios::out` and `ios::app` are three types of mode to open a file. `ios::in` and `ios::out` mean the files are of read and write states, respectively. In `ios::out` mode, the stream will clean out the information in an existing file and write the text in the beginning of file. However, the program writes the new text from the end of file under the `ios::app` mode.

- Here below shows the content of file `lab12-2.txt`

```
  When your program takes input from a file, it is said
to be reading from the file; when your program sends output
to a file, it is said to be writing to the file.
new words
```

Lab 12-4: Binary File

✓ In this lab, you will learn how to write and read the binary file.

```
// lab12-4-1.cpp
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    ofstream out("out.dat", ios::binary);
    for ( int i = 0 ; i < 10 ; ++i) {
        out.write((char*)&i, sizeof(i));
    }
    out.close();
    return 0;
}
```

- Use member function `write( (char*)&var , sizeof(var) )` to write a binary file.

```
// lab12-4-2.cpp
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    int num = 0;
    ifstream in("out.dat");
    for ( int i = 0 ; i < 10 ; ++i) {
        in.read((char*)&num, sizeof(i));
        cout << num << endl;
    }
    in.close();
    return 0;
}
```

- Use member function `read( (char*)&var , sizeof(var) )` to read a binary file..

Exercise 12-1

✓ Please write a C/C++ program to calculate the arithmetic expression including addition (+) and subtraction (–) operation from the command line argument. The corresponding syntax will look like “`ex12-1 <arith_expr>`” where `<arith_expr>` is an arithmetic expression and the numbers and the operators in such an expression will be separated by one blank space. For convenience, the leading token will be always one number (either positive or negative). Two examples of the required format are given as follows.

```
>./ex12-1 12 + 3 – 7 + 45 - 27
Answer is: 26
>
>./ex12-1 -17 + 30 - 22 + 9
Answer is: 0
>
```

✓ Please write a C/C++ program to implement a function to search a target word and count the frequency of its appearance in one file. The target word and the filename are both read from the command line. Given the input file `ex12-2.txt`, each row contains one long string.

```
abdaaabdabceepowpabcpwpw
dsdskldksklwasslwlasdklwwpaoabcqqq
dwdwlwppowknmwi
```

- Two examples of the required format are given as follows

```
>./ex12-2 ex12-2.txt abc
"abc" appears 3 time(s) in ex12-2.txt.
>
>./ex12-2 ex12-2.txt bb
"bb" appears 0 time(s) in ex12-2.txt.
>
```