UEE1302(1066) F12: Introduction to Computers and Programming

Lab 10: C-String & Class String & File I/O

What you will learn from Lab 10

In this laboratory, you will understand how to use C++ ***class strings*** and learn the usage of file I/O.

TASK 10-1 : FILE INPUT STREAM

- ✓ Program `lab10-1` shows an example of using file input stream to read data from the text file `lab10-1.txt`.
- ✓ The below shows the context of file `lab10-1.txt`.

```
1 4.5 file
```

- ✓ Program `lab10-1` is listed below. To use `ifstream` to read text file, `<fstream>` library is required to be included.

```
// file: lab10-1.cpp
#include <fstream>
#include <iostream>
using namespace std;

int main()
{
    ifstream in_file;
    in_file.open("lab10-1.txt");

    int vali;
    double vald;
    char valc[50];

    in_file >> vali >> vald >> valc;

    cout << "Integer number in file: " << vali << endl;
    cout << "Floating number in file: " << vald << endl;
    cout << "Character in file: " << valc << endl;

    in_file.close();
    return 0;
}
```

  - ➢ `in_file` is declared as an input stream variable. `open()` is one of member function of `in_file` to open the text file and `close()` is used to close the text file.
  - ➢ Use the extraction operator >> to take input from the file

TASK 10-2: FILE OUTPUT STREAM

✓ Program `lab10-2` gives an example to output the results to file `lab10-2.txt`. Compare with `ifstream` to read file, an output stream `ofstream` is used to write to the file

```
// file: lab10-2.cpp
#include <fstream>
#include <iostream>
using namespace std;

int main()
{
   char *filename;
   cout << “Enter the filename:”;
   cin >> filename;

   ofstream out_file(filename);

   int n;
   cout << "Enter an number:";
   cin >> n;
   for (int idx = 0; idx < n; idx++)
   {
      out_file << idx << endl;
   }

   cout << "Write file lab10-2.txt done!" << endl;
   out_file.close();

   return 0;
}
```

✓ Below shows the result of executing program `lab10-2`.

```
> ./lab10-2
Enter the filename: lab10-2.txt↵
Enter an number: 5↵
Write file lab10-2.txt done!
> cat lab10-2.txt
0
1
2
3
4
```

TASK 10-3 : FILE INPUT STREAM: IFSTREAM

- ✓ Program `lab10-3` provides an example of using for file input stream to read information from the file `lab10-3.txt`.
- ✓ Here below shows the content of file `lab10-3.txt`.

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

- ✓ Program `lab10-3` asks the user to input the filename to open and check if the file can be opened successfully.

```
//File: lab10-3.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.

- ✓ Program `lab10-4`gives an example to read the text file `lab10-4.txt`.

✓ Here below shows the content of file `lab10-4.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 `lab10-4` uses a different method to open the file.

```
// file: lab10-4.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 `./lab10-4 lab10-4.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.

TASK 10-4: FILE OUTPUT STREAM: OFSTREAM

✓ Program `lab10-5` gives an example to append text in an existing file `lab10-4.txt`.

```
// file: lab10-5.cpp
#include <fstream>
#include <iostream>
using namespace std;
```

```
int main()
{
    ofstream myfile;
    myfile.open("lab10-4.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 `lab10-4.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
```

TASK 10-5: EXERCISES

✓ Please write a program to extend ex9-1. The new program can read a input file and write a output file. The result in output file which has counts the number of words in each sentence and displays words one-by-one.

- Here below shows the content of file `ex10-1.1.txt`

```
I am Charles.
I am a handsome boy.
I love C++ programing.
```

- The comment `shows as follow:`

```
> ./ex10-1 ex10-1.1.txt ex10-1.out
Done!!!
```

- The file `ex10-1.out shows as follow:`

```
line 1 has 3 words, "I", "am", "Charles"
line 2 has 5 words, "I", "am ", "a", "handsome", "boy"
line 3 has 4 words, "I", "love", "C++", "programing"
```

✓ Typesetting problem

The input file is a line without formal typesetting. Please write a program to restructure the input file, and print out the result in output file.

The output file follows these rules

1. If met punctuation "**,**", you should add a space after "**,** ".
2. If met punctuation "**.**", you should add a "**n**" after "**.**".
3. The number of characters in each line is smaller than **60**.

➢ Here below shows the content of file `ex10-2.1.txt`

```
It's not the hot commodity it was last year but gold has been creeping its
way back to prominence of late.The yellow metal is once again over $1,700 an
ounce, up more than 10% for the year,and well off the lows hit early last
summer.With gold in the upper end of the $1,500 to $1,900 range that it has
traded in since making all-time highs in 2011,it's a good time for investors
to question whether it's time to load up ahead of a potential push to new highs.
```

➢ The comment shows as follow:

```
> ./ex10-2 ex10-2.1.txt ex10-2.out
Done!!!
```

➢ The file `ex10-2.out` shows as follow:

```
It's not the hot commodity it was last year but gold has
been creeping its way back to prominence of late.
The yellow metal is once again over $1,700 an ounce, up
more than 10% for the year, and well off the lows hit early
last summer.
With gold in the upper end of the $1,500 to $1,900 range
that it has traded in since making all-time highs in 2011,
it's a good time for investors to question whether it's
time to load up ahead of a potential push to new highs.
```

Exercise after LAB

1. Most Frequent Character

Write a function that accepts either a pointer to a C-String, or a string object, as its argument. The function should return the character that appears most frequently in the string. Demonstrate the function in a complete program.

2. ReplaceSubstring Funttion

Write a function named replaceSubstring. The function should accept three C-sting or string object arguments. Let's call them string1, string2, and string3. It should search string1 for all occurrences of string2. When it finds an occurrence of string2, it should replace it with string3. For example, suppose the three arguments have the following value:

string1: **"the dog jumped over the fence"**
string2: **"the"**
string3: **"that"**

with these three arguments, the function would return a string object with the value **"that dog jumped over the fence"**. Demonstrate the function in a complete program.

3. User Name

Write a program that queries its environment, determines the user's login name, and then greets the user by name. For example, if the login name of the user is **gcm**, then the program prints

Hello, gcm

When it is executed