GECG10069 (561085) F25: Introduction to Programming (C++)

Lab 10: File I/O & Stringstream

What you will learn from Lab 10

In this laboratory, you will review how to use file I/O and understand how to use stringstream.

TASK 10-1: FSTREAM AND ARGV

✔ Program `lab10-1` gives an example of using `fstream` to open file.

```
// File: lab10-1.cpp
#include <fstream>
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    fstream out("input.txt", ios::in | ios::out);
    if (out.fail())
    {
        cout << "Cannot open the file " << argv[1] << endl;
        return 1;
    }

    String line;
    out ≫ line;
    cout ≪ line;
    return 0;
}
```

➢ `ios::in | ios::out` means that the file can be operated in two modes.

➢ `return 1` indicates that the program terminates in a non-normal way. `exit(1)` is the same as `return 1`, but `exit(1)` can be used in the any return type of main function. For example, `exit(1)` can be used in `void main()`, but `return 1` cannot.

✔ Executing program `lab10-1`

```
// input.txt
When our 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 10-2: FILE INPUT AND OUTPUT STREAM: SSTREAM

✔ Program `lab10-2` gives an example to using `sstream`, which can use the read, write, and pass data.

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

```
int main()
{
    stringstream ss;
    int data = 200;
    int result;
    ss << data;
    ss >> result;
    cout << result << endl;

    return 0;
}
```

```
// File: lab10-3.cpp
#include <iostream>
#include <string>
#include <sstream>
using namespace std;
int main()
{
    stringstream ss;
    string name = "Tom";
    ss << "Hello " << name <<endl;
    cout << ss.str();

    return 0;
}
```

- The "`.str()`" returns a string object with a copy of the current contents of the stream.

```
// File: lab10-4.cpp
#include <iostream>
#include <string>
#include <sstream>
using namespace std;
int main()
{
    stringstream ss;
    ss.str("lab10-4.cpp example");
    string s = ss.str();
    cout << s <<endl;

    return 0;
}
```

- Assign `stringstream ss`

✔

```
// File: lab10-5.cpp
#include <iostream>
#include <string>
#include <sstream>
using namespace std;
int main()
{
    stringstream ss;
    string strvalues = "32 240 2 145";
    ss.str(strvalues);
    for(int n=0 ; n < 4 ; n++) {
        int val;
        ss >> val;
        cout << val <<endl;
    }
    cout << "Finished writing the numbers in:" <<endl;
    cout << ss.str() << endl;
    return 0;
}
```

- string to int by using `stringstream`

Exercise 10-1 : String Length and Character Output

Description -

Read a CSV file named **10_1.csv** that records daily transactions.
Each line contains two values:

- an **item name**, and
- the **amount of money** (positive for income, negative for expense).

After reading all records, calculate and print the **final balance** in the bank account.

(Positive numbers represent income, negative numbers represent expenses.)

Input :

- A file named account.csv located in the same directory as the program.
- Each line follows the format:
  Item,Amount

Output Format :

- Final Balance: <calculated_amount>

Sample Test Cases -

Sample Input - 1

```
Salary,30000
Lunch,-120
Coffee,-60
Book,-500
Bonus,2000
```

Sample Output - 1

```
Final Balance: 31320
```

EXERCISE 10-2 : CHARACTER FREQUENCY ANALYSIS

Description -

Read a text file named 10_2.txt and count how many times each English letter (A–Z or a–z) appears.

Letters should be treated **case-insensitively** (**‘A’ and ‘a’ are the same**).

Ignore any non-alphabetic characters such as spaces, numbers, or punctuation marks.

After counting, print the top 5 most frequent letters. If multiple letters have the **same frequency**, output the letter with the **smaller alphabetical order first** (A before B, etc.).

Input Format :

- A text file named input.txt located in the same directory as the program.
- Each line may contain letters, numbers, spaces, or punctuation.

Output Format :

- Show the top 5 most frequent letters, in the format

  Top <rank>: <letter> (<count>)

Hint:

- isalpha(a) → returns true if a is an alphabetic character, otherwise returns false.
- toupper(a) → converts character a to its uppercase form.
- #include <cctype> // for isalpha, toupper
- 'B' can be represented as 'A' + 1, and so on for other letters.

Sample Testcases -

Sample Input - 1

```
Hello world!
This is a simple test.
HELLO again.
```

Sample Output - 1

```
Top 1: L (6)
Top 2: E (4)
Top 3: A (3)
Top 4: I (3)
Top 5: O (3)
```

code skeleton

```
#include <iostream>
#include <fstream>
#include <cctype>  // for isalpha, toupper
using namespace std;

int main() {
    // === TO-DO 1 : READ FILE ===


    int freq[26] = {0};
    char ch;

     // === TO-DO 2：統計每個字母出現次數 ===
     while ( /* TODO: 讀取字元 */) {
        if (isalpha(ch)) {
                ch = toupper(ch); //轉大寫
                // TODO: 將 ch 對應到 freq 陣列
        }
    }


    char letters[26];
    for (int i = 0; i < 26; i++) {
        letters[i] = 'A' + i;
    }

     // === TO-DO 3：排序 (依頻率高→低, 字母小→大) ===
    for (int i = 0; i < 25; i++) {
        for (int j = i + 1; j < 26; j++) {

            // TODO: 若 freq[j] > freq[i], 或次數相同但 letters[j] < letters[i]
            //            則交換 freq[i] ↔ freq[j], letters[i] ↔ letters[j]

        }
    }

     // 輸出前 5 名
    cout << "Top 5 letters:" << endl;
    for (int i = 0; i < 5; i++) {
        cout << "Top " << i + 1 << ": " << letters[i]
             << " (" << freq[i] << ")" << endl;
    }

    return 0;
}
```