Basics: From C to C++

Computer Programming for Engineers (DSAF003-42)
Fall, 2021

Practice 4 : String-io

Instructor:

Youngjoong Ko (nlp.skku.edu)

String append

■ To append string use '+' operator or 'append'

```cpp
1    #include <iostream>
2    #include <string>
3    using namespace std;
4
5    int main( )
6    {
7        string lastname;
8        string firstname;
9        cout << "Type your lastname: ";
10       cin >> lastname;
11       cout << "Type your firstname: ";
12       cin >> firstname;
13
14       cout << "My name is " << firstname + " " + lastname << endl;
15       cout << "My name is " << firstname.append(" ").append(lastname) << endl;
16   }
```

```
Type your lastname: huh
Type your firstname: taehun
My name is taehun huh
My name is taehun huh
```

String cin, comparison

- cin use whitespace as delimiter and can't get a complete line
- String can compare each other

```cpp
1    #include <iostream>
2    #include <string>
3    using namespace std;
4
5    int main( )
6    {
7        string username;
8        string pass, checkpass;
9        cout << "Type name and password: ";
10       cin >> username >> pass;
11       cout << "Type password again: ";
12       cin >> checkpass;
13
14       if(pass == checkpass) cout << "Login Successfully\n";
15       else cout << "Wrong Password\n";
16       return 0;
17   }
```

```
Type name and password: taehun qwer123
Type password again: qwer123
Login Successfully
```

```
Type name and password: taehun qwer123
Type password again: asdf123
Wrong Password
```

getline() Example

- getline() without delimiter can get a complete line
- getline() with delimiter read until delimiter and strings after delimiter remains in cin

```cpp
1    #include <iostream>
2    #include <string>
3    using namespace std;
4
5    int main( )
6    {
7        string line;
8
9        cout<<"Enter input: "; getline(cin,line);
10       cout << "line: " << line << endl;
11
12       cout<<"Enter input: "; getline(cin,line,'?');
13       cout << "line: " << line << endl;
14
15       //strings after delimiter remains in cin
16       getline(cin, line);
17       cout << "line: " << line << endl;
18       return 0;
19   }
```

```
Enter input: hello? world
line: hello? world
Enter input: hello? world
line: hello
line:  world
```

Exercise 1

■ **Fill main code and follow rules below**

- get a student ID and one line as input

- A line input must be name:score

    ex) taehun:90

- Output format is <name>(<student ID>)'s score is <score>

output example

```
Type student ID: 2021711919
Type name and score: taehun:90

taehun(2021711919)'s score is 90
```

String member functions

■ This is just example for some functions, you can find more in c++ document

```cpp
1   #include <iostream>
2   #include <string>
3   using namespace std;
4
5   int main()
6   {
7       string str;
8       if (str.empty())
9           cout << "str is empty!\n";
10
11      str = "welcome! cpp";
12      cout << "Length of str: " << str.length() << endl;
13      cout << "4'th element of std: " << str.at(3) << endl;
14      cout << "5'th element of std: " << str[4] << endl;
15      cout << "'!'is " << str.find('!')+1 << "'th element\n";
16
17      string str2(str.substr(0,5));
18      cout << "str2: " << str2 << endl;
19      str2.append("!");
20      cout << "str2 after append: " << str2 << endl;
21  }
```

```
str is empty!
Length of str: 12
4'th element of std: c
5'th element of std: o
'!'is 8'th element
str2: welco
str2 after append: welco!
```

Output formatting example

```cpp
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string str("Hello");
    double pi = 3.141592;
    cout.width(10);
    cout << str <<endl;

    cout.width(10);
    cout.fill('*');
    cout << str <<endl;

    cout.width(10);
    cout.fill('*');
    cout.setf(ios::left);

    cout << str << endl;
    cout << pi << endl;

    cout.setf(ios::fixed);
    cout.precision(2);
    cout << pi << endl;
    return 0;
}
```

```
     Hello
*****Hello
Hello*****
3.14159
3.14
```

String conversion example

- Class string can be converted to numeric type and c-style string
- c_str() returns a char array(c-style string)
- stoi and to_sting are conversion function

```cpp
1    #include <iostream>
2    #include <string>
3    #include <cstring>
4    using namespace std;
5
6    int main()
7    {
8        string str;
9        char c_str[] = "hello";
10       str = c_str;
11       cout << str << endl;
12
13       str = "52'th";
14       cout << c_str << endl;
15       strcpy(c_str, str.c_str());
16       cout << c_str << endl;
17
18       int i = stoi(str); i++;
19       str = to_string(i) + "'th";
20       cout << str << endl;
21       return 0;
22   }
```

```
hello
hello
52'th
53'th
```

Exercise 2

- **Write the program that described below**
  - Get a line as input
  - Print reverse string of input string

output example

```
Input String: I will be back
Reversed String: kcab eb lliw I
```

Assignment

- **Write the program that described below**
  - Get one line as input
  - Input format : name phone number
  - White space(between name and phone number) is delimiter
  - Print inputted name
  - Print phone number that is masked to '*' except area code of phone number

<div style="text-align:center">output example   output example</div>

```
Input name and phone number: taehun 010-3262-4758
Name: taehun
Masked Phone Number: 010-****-****
```

```
Input name and phone number: SKKU 02-760-0114
Name: SKKU
Masked Phone Number: 02-***-****
```