【C++】Day nine

Class	C++
Date	@November 27, 2021
Material	
Series Number	
Summary	Subscript Operator and Iteration

【Ch3】Strings(3)

Use a Range for to Change the Characters in a string

If we want to change the value of the characters in a string, we must define the loop variable as a reference type. A referenece is just another name for a given object. When we use a reference as our control variable, that variable is bound to each element in the sequence in turn.

For example, if we want to turn all of the letters in a string to uppercase: (using the library function `toupper`)

```
string str("Hello World");
for(auto &c : str)
  c = toupper(c);
```

The output of this code is "HELLO WORLD"

Processing Only Some Characters

There are two ways to access individual characters in a string: We can use a subscript or an iterator. We will discuss the subscript here and talk iterator later.

The subscript operator `[]` takes a `string::size_type` value that dentoes the position of the character we want to acess. The operator returns a reference to the character at the given position.

The first element in the string is `s[0]` and the last element is `s[s.size() - 1]`.

Note: The values we use to subscript a string must be ≥ 0 and < size(). The result of using an index outside this range is undefined. By implication, subscripting an empty string is undefined.

Using a Subscript for Iteration

As another example, we will change the first word in a string to uppercase

```
for(decltype(s.size()) index = 0; index < s.size() && !isspace(s[index]); ++index)
  s[index] = toupper(s[index]); //capitalize the current character
```

Using a Subscript for Random Access

Let's assume we have a number between 0 and 15 and we want to generate the hexademical representation of that number. See the following code for an solution

```
const string hex = "0123456789ABCDEF";
string result; //hold the resulting hexified string
string::size_type n;
while(cin >> n)
  if(n < 15)
    result += hex[n];
cout << "Your hex number is " << result << endl;
```