# Lecture 13: Array and Vector – Part III

Class page: https://github.com/tsung-wei-huang/cs1410-40

Dr. Tsung-Wei Huang
Department of Electrical and Computer Engineering
University of Utah, Salt Lake City, UT

# String

- **The string in C/C++ is a series of characters that are treated as a single unit**
  - Written in double quotes, ex: “Hello”
- **In C, strings are arrays of characters that end with an extra character '\0‘ (NULL, ASCII code=0)**
  - Ex: ‘H’, ‘e’, ‘l’, ‘l’, ‘o’, ‘\0’ (6 elements)
  - Not convenient to deal with
- **C++ standard library class `string` provides more powerful capabilities**

# String

```
1
2  // Standard Library class string.
3  #include <iostream>
4  #include <string>
5  using namespace std;
6
7  int main()
8  {
9     string s1( "happy" ); // string object s1 initialized to "happy"
10    string s2( " birthday" ); // s2 initialized to " birthday"
11    string s3; // s3 is empty
12    string s4; // s4 is empty
13
14    // read a line of text into a string object
15    cout << "Enter a line of text: ";
16    getline( cin, s4 ); // read line of text into s4
17
18    // display each string's contents
19    cout << "s1 is \"" << s1 << "\"; s2 is \"" << s2
20       << "\"; s3 is \"" << s3 << "\"; s4 is \"" << s4 << "\"\n\n";
21
22    // determine the length of each string
23    cout << "s1 length " << s1.length() << "; s2 length " << s2.length()
24       << "; s3 length " << s3.length() << "; s4 length " << s4.length();
```

# String

```
25
26      // test equality and relational operators
27      cout << "\n\nThe results of comparing s2 and s1:"
28         << "\ns2 == s1 yields " << ( s2 == s1 ? "true" : "false" )
29         << "\ns2 != s1 yields " << ( s2 != s1 ? "true" : "false" )
30         << "\ns2 >  s1 yields " << ( s2 > s1 ? "true" : "false" )
31         << "\ns2 <  s1 yields " << ( s2 < s1 ? "true" : "false" )
32         << "\ns2 >= s1 yields " << ( s2 >= s1 ? "true" : "false" )
33         << "\ns2 <= s1 yields " << ( s2 <= s1 ? "true" : "false" );
34
35      // test string member function empty
36      cout << "\n\nTesting s3.empty():" << endl;
37
38      if ( s3.empty() )
39      {
40         cout << "s3 is empty; assigning s1 to s3;" << endl;
41         s3 = s1; // assign s1 to s3
42         cout << "s3 is \"" << s3 << "\"";
43      } // end if
44
45      // test string concatenation operator
46      cout << "\n\nAfter s1 += s2, s1 is ";
47      s1 += s2; // concatenate s2 to the end of s1
48      cout << s1;
```

# String

```
49
50      // test string concatenation operator with string literal
51      cout << "\n\ns1 += \" to you\" yields" << endl;
52      s1 += " to you";
53      cout << "s1 is " << s1 << "\n\n";
54
55      // test string member function substr
56      cout << "The substring of s1 starting at location 0 for\n"
57         << "14 characters, s1.substr(0, 14), is:\n"
58         << s1.substr( 0, 14 ) << "\n\n"; // displays "happy birthday "
59
60      // test substr "to-end-of-string" option
61      cout << "The substring of s1 starting at\n"
62         << "location 15, s1.substr(15), is:\n"
63         << s1.substr( 15 ) << endl; // displays "to you"
64
65      // making a copy of a string
66      string s5( s1 ); // creates s5 as a copy of s1
67      cout << "\ns5 is " << s5;
68
69      // test the subscript operator to create lvalues
70      s1[ 0 ] = 'H'; // replaces h with H
71      s1[ 6 ] = 'B'; // replaces b with B
72      cout << "\n\ns1 after s1[0] = 'H' and s1[6] = 'B' is: " << s1;
```

# String

```
73
74       // test the subscript operator to create rvalues
75       cout << "\n\ns1[0] is " << s1[ 0 ] << "; s1[2] is " << s1 [ 2 ]
76          << "; s1[s1.length()-1] is " << s1[ s1.length() - 1 ];
77
78       // test subscript out of range with string member function "at"
79       cout << "\n\nAttempt to assign 'd' to s1.at( 30 ) yields:" << endl;
80       s1.at( 30 ) = 'd'; // ERROR: subscript out of range
81    } // end main
```

```
Enter a line of text: Using class string
s1 is "happy"; s2 is " birthday"; s3 is ""; s4 is "Using class string"

s1 length 5; s2 length 9; s3 length 0; s4 length 18

The results of comparing s2 and s1:
s2 == s1 yields false
s2 != s1 yields true
s2 >  s1 yields false
s2 <  s1 yields true
s2 >= s1 yields false
s2 <= s1 yields true
```

# String

```
Testing s3.empty():
s3 is empty; assigning s1 to s3;
s3 is "happy"

After s1 += s2, s1 is happy birthday

s1 += " to you" yields
s1 is happy birthday to you

The substring of s1 starting at location 0 for
14 characters, s1.substr(0, 14), is:
happy birthday

The substring of s1 starting at
location 15, s1.substr(15), is:
to you

s5 is happy birthday to you

s1 after s1[0] = 'H' and s1[6] = 'B' is: Happy Birthday to you

s1[0] is H; s1[2] is p; s1[s1.length()-1] is u

Attempt to assign 'd' to s1.at( 30 ) yields:
```
*Platform specific error message will be dislayed*

# Example: Count Frequency

- **Given a string s, count the frequency of alphabets**
- **Example: afeqfasc**
- **Output:**

a: 1

b: 0

c: 1

…

f: 2

…

s: 1

# Summary

- **Array**
- **Vector**
- **String**
  - One of the most commonly used data structures
  - Can represent “human-readable data”