# EECE.3220: Data Structures

Spring 2017

Lecture 26: Key Questions
April 3, 2017

1. Explain how the comparison operators (==, !=, <, >, <=, >=) can be used to compare string objects.

2. Explain how string concatenation works in C++.

3. Explain the operation of the `substr()` function.

4. Explain how to access individual characters within a string.

5. List the output for each of the following code snippets from the same program.

```
int main()
{
   string s1( "happy" );
   string s2( " birthday" );
   string s3;
   // test overloaded equality and relational operators
   cout << "s1 is \"" << s1 << "\"; s2 is \"" << s2
      << "\"; s3 is \"" << s3 << '\"'
      << "\n\nThe results of comparing s2 and s1:"
      << "\ns2 == s1 yields " << ( s2 == s1 ? "true" : "false" )
      << "\ns2 != s1 yields " << ( s2 != s1 ? "true" : "false" )
      << "\ns2 >  s1 yields " << ( s2 > s1 ? "true" : "false" )
      << "\ns2 <  s1 yields " << ( s2 < s1 ? "true" : "false" )
      << "\ns2 >= s1 yields " << ( s2 >= s1 ? "true" : "false" )
      << "\ns2 <= s1 yields " << ( s2 <= s1 ? "true" : "false" );
```

***OUTPUT:***

```
// test string member function empty

   cout << "\n\nTesting s3.empty():" << endl;
   if ( s3.empty() )
   {
      cout << "s3 is empty; assigning s1 to s3;" << endl;
      s3 = s1; // assign s1 to s3
      cout << "s3 is \"" << s3 << "\"";
   } // end if
   // test overloaded string concatenation operator
   cout << "\n\ns1 += s2 yields s1 = ";
   s1 += s2; // test overloaded concatenation
   cout << s1;
   // test concatenation operator with C-style string
   cout << "\n\ns1 += \" to you\" yields" << endl;
   s1 += " to you";
   cout << "s1 = " << s1 << "\n\n";
```

***OUTPUT:***

```
// test string member function substr

   cout << "The substring of s1 starting at location 0 for\n"
      << "14 characters, s1.substr(0, 14), is:\n"
      << s1.substr( 0, 14 ) << "\n\n";
   // test substr "to-end-of-string" option
   cout << "The substring of s1 starting at\n"
      << "location 15, s1.substr(15), is:\n"
      << s1.substr( 15 ) << endl;
   // test using subscript operator to create lvalue
   s1[ 0 ] = 'H';
   s1[ 6 ] = 'B';
   cout << "\ns1 after s1[0] = 'H' and s1[6] = 'B' is: "
      << s1 << "\n\n";
   // test subscript out of range with string member function "at"
   cout << "Attempt to assign 'd' to s1.at( 30 ) yields:" << endl;
   s1.at( 30 ) = 'd'; // ERROR: subscript out of range
   return 0;
} // end main
```

***OUTPUT:***