String(s) and Char(s)

Instructor: Andy Abreu
Strings

• String is a class built into the C++ library.

• It is there to place the original “cstrings” concept first developed for C.

• String has predefined functions contained within the class which we can use for our convenience to do string manipulations.
Declaring a String

- Model:
  
  ```
  string variable_name;
  ```

- string firstname = "Tyler";
- string lastname = "Tseng";
Reading Into a String
Reading a word into a string

• To read in a person’s first name, we can do the following:

```cpp
string firstname;
cout << "Enter your first name: ";
cin >> firstname;
cout << "Your are " << firstname;
```

• You might note using cin like this only reads in one word at a time. Cin will read to the next whitespace.
Reading a line into a string

• To read in a person’s full name, we can do the following:

    string fullname;
    cout << "Enter your full name: ";
    getline( cin, fullname );
    cout << "Your are " << fullname;

• Getline function will read in all the characters entered until it hits the newline character
String as Char array
Parts of the string

• We can think of string as an array of characters
• So a string defined as the follows...
  string name = "Tyler";
• Could be thought of as...

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>T</td>
<td>y</td>
<td>l</td>
<td>e</td>
<td>r</td>
</tr>
</tbody>
</table>

• Where...
• name[0] = 'T'; name[1] = 'y';
Replacing parts of the string

• Following the previous example, if we had the following code...

    name[1] = 'i';
    name[2] = 'g';

• This would turn Tyler Tiger
String Manipulation
Identify the length of a string

• To identify the length of the string, we can use one of the following string class methods:

```cpp
string str = "today is my birthday";
cout << "length: " << str.length();
//or
cout << "length: " << str.size();
```
Concatenate two strings together

• To concatenate two strings together, we can do the following:

```cpp
string s1 = "Hello ";
string s2 = "World";
string s3 = s1 + s2;
cout << s3 << endl;
```

• Programs design with multi-lingual specifications uses a language file to hold all the text. Text is load into a variable, and the variable is cout to the screen.
Inserting a into a string

• The string library also allows us to insert some text into part of the string. We can use the insert function to do this.

• Model:
  – string_variable.insert( index_position, text_to_insert );
    • string_variable: a variable declared as a string type
    • insert: the insert function
    • index_position: the position you want the text to go this would push all other text back
    • text_to_insert: the text you want to insert in to this string
Inserting a into string (Example)

• To insert a string into another, we can do the following:
  
  string str = "NY";
  str.insert( 1, "ew " );

  //insert into the end
  str.insert( str.size(), "ork" );

  //note the location is 5
  cout << str << endl;
Comparison on strings

- Comparison on the strings are done on individual character’s code known as the ASCII code.

- Following comparisons operator are defined:
  
  ```
  ==
  true if str1 == str2

  >
  true if str1 > str2

  <
  true if str1 < str2
  ```

  ```
  !=
  true if str1 != str2

  >=
  true if str1 >= str2

  <=
  true if str1 <= str2
  ```

- Because the above operators are defined, we can sort strings in C++ as we can sort numbers.