string(s) and char(s)

Instructor: Krishna Mahavadi
Strings

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

• It is there to replace 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:

```c
string variable_name;
```

• string firstname = "Krishna";

• string lastname = "M";
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:

```cpp
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 is defined as follows...
  ```
  string name = "Krishna";
  ```
- Could be thought of as...

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>R</td>
<td>I</td>
<td>S</td>
<td>H</td>
<td>N</td>
<td>A</td>
</tr>
</tbody>
</table>

- Where...
- `name[0] = 'K'; name[1] = 'R';`
Replacing parts of the string

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

```java
name[4] = 't';
name[5] = 'e';
```

• This would turn Krishna Kristen
String Manipulation
Identify the length of a string

• To identify the length of a 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 we can use +.

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

This prints Hello World! to the screen.
• Programs designed with multi-lingual specifications use a language file to hold all the text. Text is loaded into a variable, and the variable is cout to the screen.
Inserting a string 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:

  
<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>==</td>
<td>true if str1 == str2</td>
</tr>
<tr>
<td>&gt;</td>
<td>true if str1 &gt; str2</td>
</tr>
<tr>
<td>&lt;</td>
<td>true if str1 &gt; str2</td>
</tr>
<tr>
<td>!=</td>
<td>true if str1 != str2</td>
</tr>
<tr>
<td>&gt;=</td>
<td>true if str1 &gt;= str2</td>
</tr>
<tr>
<td>&lt;=</td>
<td>true if str1 &lt;= str2</td>
</tr>
</tbody>
</table>

• Because the above operators are defined, we can sort strings in C++ as we can sort numbers.
Other String Methods:

1. `str.insert(pos, str2)` Inserts `str2` before the position `pos` of string `str`.

   Example:
   ```
   string str = “NY”;
   string str2 = “ew “;
   string str3 = “ork”;
   str.insert(1, str2);
   str.insert(5, str3);
   ```
2. `str.find(str2)` finds the `str2` in string `str`.
   Example:
   ```cpp
   string str = "There is a needle in the haystack.";
   string str2 = "needle";
   int found = str.find(str2);
   if(found >= 0)
       cout << "needle is found at: " << found << '
';
   string str3 = "haystack";
   found = str.find(str3);
   cout <<""haystack is found at: "<<found<<'\n';
   ```
3. `str.rfind(key)` finds the occurrence of string key in string str.

```cpp
string str = "A politician is an animal who can sit on a fence and yet keep both ears to the ground."
string key = "ears"
int found = str.rfind(key);
cout << found << endl;
```
4. `str.substr(pos, len)` constructs a substring of length `len`, at position `pos` of the string `str`.

```cpp
string str = "A column of smoke rose thin and straight from the chimney.";
string str2 = str.substr(12, 5);
cout << str2 << endl; // prints smoke.
```