



STRINGS

CS A150 – C++ Programming 1

STRINGS

- Sequence of characters
- String literals are enclosed in double-quotes

```
"Hello"
```

- Need to include the string header file

```
#include<string>
```

- Declared and assigned much like numbers

```
string name = "Mary"; // declaration and
                      //      initialization

name = "Jill";        // assignment
```

STRINGS (CONT.)

- **cin >>** Reads one word up to the next white space or newline character (the **[Enter]** key)

```
cin >> someVariable;
```

- **getline()** Reads all **keystrokes** until the newline character (the **[Enter]** key)

```
getline(cin, name);
```

- The newline is extracted from the stream, but **not** stored with the string
- If the user enters

Harry Hacker

then **name** will store the string **"Harry Hacker"**

STRINGS (CONT.)

- The number of characters in a string is its **length**.
- You can find a string's length by using the **member function `length()`**
- Invoked using **dot notation**

```
string name = "Bob";  
cout << name.length();
```

will output 3.

SUBSTRINGS

- The **substr** member function is used to extract substrings

```
s.substr(start, length)
```

- **start** → the start index to read the characters
- **length** → the number of characters to be read
- **Indices are zero-based** (first character is at index 0)
- The last character is always one less than the length of the string.

SUBSTRINGS

```
string greeting = "Hello, World!\n";  
string sub = greeting.substr(0, 4);
```

H	e	l	l	o	,		W	o	r	l	d	!	\n
0	1	2	3	4	5	6	7	8	9	10	11	12	13

```
string w = greeting.substr(7, 5);
```

H	e	l	l	o	,		W	o	r	l	d	!	\n
0	1	2	3	4	5	6	7	8	9	10	11	12	13

STRING FUNCTIONS

Name	Purpose
<code>s.length()</code>	The length of s
<code>s.substr(i)</code>	The substring of s from index i to the <i>end</i> of the string
<code>s.substr(i, n)</code>	The substring of s of length n starting at index i
<code>getline(f, s)</code>	Read string s from the input stream f

CONCATENATION

- The **+** operator concatenates two strings

```
string firstName = "Harry";  
string lastName = "Hacker";  
string name = firstName + " " + lastName;
```

This will become the new string name “Harry Hacker”.

CHARACTERS AND C STRINGS

- **char** data type stores a single character
 - A character literal is denoted by single quotes:

```
'c' 'Z' '\n' '\t' '\\'
```

- "C strings" are arrays of **chars**
 - Require a lot of work for the programmer
 - The last character is always the **NULL** character `'\0'`

```
char myString[ ] = "Do Re Mi";
```

'D'	'o'		'R'	'e'		'M'	'i'	'\0'
0	1	2	3	4	5	6	7	8

The left side of the slide features a series of vertical stripes in various shades of brown, tan, and grey. Overlaid on these stripes are several orange circles of different sizes. One large circle is positioned near the top left, while several smaller circles are scattered below it, some overlapping the stripes.

10

Strings (end)