CPS235 Object Oriented Programming in C++

LAB6 BESE-15A *19th April 2010*

The string class in C++

Objectives

By the end of this lab, you should be able to declare strings in C++ and use various library functions to manipulate the strings.

The Standard C++ String Class

We can represent text in C++ using objects of a 'string' class. In order to use this class you should include the following statements:

```
#include <string>
```

A string can now be created as follows.

```
string s = "abc def abc";
string s2 = "abcde uvwxyz";
```

Commonly used methods of the string class

The following table includes methods for reading string input from the stdin and commonly used methods with strings. The variable 'c', used below if of type `char` and the variable i is of type `int`.

Input	`cin >> s;`	The value of s is read from the stdin. The value read stops at whitespace. If "charlie brown" is entered in response to `cin>>s`, only "charlie" gets stored in s.
Output	`cout <<s;`	Writes the string to the specified output stream.
Line input	`getline(cin, s);`	Reads everything up to the next newline character and puts the result into the

		specified string variable.

Assignment	`s = s2;`
	`s = "abc";`
		A string literal or a string variable can be assigned to a string variable.

Subscript	`s[1] = 'c';`
	`c = s[1];`
		Changes s to equal "acc def abc" Sets c to 'b'. The subscript operator returns a char value, not a string value.

Length	`i = s.length();`	i is set to the current length of the string s

Empty	`if(s.empty())`
	`i++;`
		The example adds 1 to i if string s is empty

Relational operators	`if (s < s2)`
	`i++;`
		Uses ASCII code to determine which string is smaller. Here the condition is true because a space comes before letter d

Concatenation	`s2 = s2 + "x";`
	`s2 += "x";`
		Both examples add x to the end of s2

Substring	`s = s2.substr(1,4);`
	`s = s2.substr(1,50);`
		The first example starts in position 1 of s2 and takes 4 characters, setting s to "bcde". In the second example, s is set to "bcde uvwxyz". If the length specified is longer than the remaining number of characters, the rest of the string is used. The first position in a string is position 0. The `substr` method doesn't change the string object on which it is called. i.e. the s2 here is not changed after the

		`substr` operation.
Substring replace	`s.replace(4,3,"x");`	Replaces the three characters of s beginning in position 4 with the character x. Variable s is set to "abc x abc". The `replace` method does change the string object on which it is called.

Substring removal	`s.erase(4,5);`
	`s.erase(4);`
		Removes the five characters starting in position 4 of s. The new value of s is "abc bc". Remove from position 4 to end of string. The new value of s is "abc ". The `erase` method does change the string object on which it is called.

Pattern matching	`i = s.find("ab",4);`	The first example returns the position of the substring "ab" starting the search in position 4. Sets i to 8. Exercise: What is returned if the substring doesn't exist?

Task for Today

Write a simple program that contains a group of functions. Each should accept a string as an argument and return a Boolean. The function should verify that string on different criteria in each function which are listed below.

Important part to note is that you have to use string manipulation functions for the verification. Do not check individual indexes by looping through the string.

If all conditions are met, function should return true, otherwise it should return false.

You may have to use google to find a complete list of functions. You might need to use the following functions as well

isalpha(char)
isdigit(char)

IsValidEmail()
Should
Have '@' (at) symbol
Have at least three characters before @
Have '.' (dot)
Should not have any space.

e.g Abdullah@gmail.com

IsValidNIC()
Should
Have exact 15 characters.
have first five characters as numeric
have sixth character as a '–' (dash)
have next 7 characters numeric
have next single character as '-'
then last character should be a numeric.
e.g: 37872-0374632-1

IsValidAccount()
Format: CC-XXXX-XXXX-XX
Where C is a character
And X is a numeric value

IsValidMob()
Should be of exactly 12 characters long
First two digits should be 00
Followed by any two numeric
Next should be '–'(dash)
Followed by any three numeric
Next should be '–' (dash)
Followed by any 7 numeric.
e-g: 0092-333-5897864

IsValidLand()
0092-51-8509855

IsValid Name()
Should only contain alphabets. No special characters or numeric characters.