

## CSC 150 LAB 2-2 MORE FUN WITH STRINGS

PROF GODFREY C. MUGANDA  
DEPT. OF COMPUTER SCIENCE  
NORTH CENTRAL COLLEGE

Here is a list of other string member functions.

### 1. STRINGS AND SINGLE CHARACTERS

In a previous lab, we saw that we can use the `append` member function to add a string to the end of another string:

```
append(string str)
```

So for example, the code

```
string str1 = "mom";  
str1.append(" and dad");  
cout << str1 << endl;
```

will print the string `mom and dad`.

1.1. **Exercise 1.** Now suppose that you ask the user to enter a string `str`, a positive integer `number`, and a character `ch`. You want to add a sequence of `n` copies of `ch` to the end of the `str` and print the resulting string. A sample interaction would look like this:

```
Enter a string: This is good  
Enter a positive integer: 5  
Enter a character: !  
The modified string is: This is good!!!!!!
```

Write the program for Exercise 1. You must use the string member function `void push_back(char ch)` that adds a single character to the end of a string. For example, if you have a string `str` and a character `ch`, then the statement

```
str.push_back(ch);
```

will add `ch` to the end of `str`.

1.2. **Exercise 2.** There is another way to solve the above problem. The string class has a member function

```
append(int n, char ch)
```

that adds `n` copies of the character `ch` to the end of a string.

To the end of the program you wrote for Exercise 1, add code that asks the user to enter another positive integer and another character. The program then adds the number of copies of the character that is specified by the integer to the end of the string already accumulated. Here is a sample run:

```
Enter a string: abcde
Enter a positive integer: 4
Enter a character: X
The modified string is: abcdeXXXXX
```

```
Enter a positive integer: 8
Enter a character: Y
The modified string is: abcdeXXXXXXXXXXXX
```

### 1.3. Exercise 3. There is also a member function

```
insert(int position, int n, char ch)
```

that can be used to insert `n` copies of the character `ch` at a given position in a string. The position is the number of character that precede the inserted copies of the character.

Using the member function, modify the program you have written so that it asks the user for position `p`, a positive integer `n` and a character `ch`, and inserts `n` copies of `ch` at the position `p`.

Here is a sample run:

```
Enter a string: abcdefg
Enter a positive integer for the repetition count: 4
Enter a character: U
The modified string is: abcdefgUUUUU
```

```
Enter a positive integer for the repetition count: 5
Enter a character: K
The modified string is: abcdefgUUUUUKKKKK
```

```
Enter a positive integer for the position: 2
Enter a positive integer for the repetition count: 6
Enter a character: h
The modified string is: abhhhhhhcdefgUUUUUKKKKK
```

## 2. A COUPLE MORE STRING CLASS FUNCTIONS

There are a couple more member functions of the string class that are useful to know.

```
push_back(char ch)
```

adds a single character to the end of a string. For example,

```
string str = "a";
str.push_back('x');
cout << str << endl;
```

prints the string "ax".

The member function

```
pop_back()
```

pops (removes) a character off of the end of a string. For example, the code

```
string str = "abcd";
str.pop_back();
cout << str << endl;
```

will print the string "abc".

### 3. EXERCISE 4

Add code that asks the user to enter a positive integer  $n$  and a character  $ch$ .

The program then uses what the user entered to print a series of geometric shapes using the entered character, as follows.

- (1) A line of  $n$  characters. Your program should do this by creating a string consisting of  $n$  characters and printing the string.
- (2) An  $n \times n$  square. Your program should do this by creating a string of  $n$  characters, and then printing the string  $n$  times using a loop.
- (3) A right triangle that is the upper left triangle of an  $n \times n$  square. Your program should do this by creating a string of  $n$  characters, printing the string, removing a character from the string, printing the string, and so on, until the string becomes empty.
- (4) A right triangle that is the lower left triangle of an  $n \times n$  square. Your program should do this by first printing a string of 1 character on a line, and then a string of 2 characters, and so on, until a string of  $n$  characters has been printed.

Here is a sample of what user interaction with the last part of your program should look like:

```
Please enter an integer: 10
Please enter a character: W
```

```
A line of 10 characters:
WWWWWWWWWW
```

```
A square of 10 by 10 characters:
WWWWWWWWWW
WWWWWWWWWW
WWWWWWWWWW
WWWWWWWWWW
WWWWWWWWWW
WWWWWWWWWW
WWWWWWWWWW
WWWWWWWWWW
WWWWWWWWWW
WWWWWWWWWW
```

```
Upper left triangle of the square:
WWWWWWWWWW
WWWWWWWWW
WWWWWWWWW
WWWWWWW
WWWWW
WWWWW
```

PROF GODFREY C. MUGANDA DEPT. OF COMPUTER SCIENCE NORTH CENTRAL COLLEGE

WWW

WWW

WW

W

Lower left triangle of the square:

W

WW

WWW

WWWW

WWWWW

WWWWWW

WWWWWWW

WWWWWWW

WWWWWWW

WWWWWWW