3.15 More string operations

Finding in a string / Getting a substring

The string library provides numerous useful functions, including functions for finding a character or string within a string, or getting a substring of a string.

Table 3.15.1: find() and substr() functions, invoked as myString.find().

```
find()
          find(item)
                           // userText is "Help me!"
                           returns index
          of first item
                           only)
                           userText.find('z')  // Returns string::npos
userText.find("me")  // Returns 5
          occurrence.
                           userText.find('e', 2) // Returns 6 (starts at index 2)
          else returns
          string::npos
          (a constant
          defined in
          the string
          library). Item
          may be char,
          string
          variable.
          string literal
          (or char
          array).
          find(item,
          indx) starts
          at index
          indx.
substr()
          substr(index,
                           // userText is "http://google.com"
                           userText.substr(0, 7)
                                                                    // Returns
          len) returns
                           "http://"
          substring
                                                                   // Returns ".com"
                           userText.substr(13, 4)
                           userText.substr(userText.size() - 4, 4) // Last 4: ".com"
          starting at
          index and
          having len
          characters.
```

Feedback?

Figure 3.15.1: Example: Get username from email address.

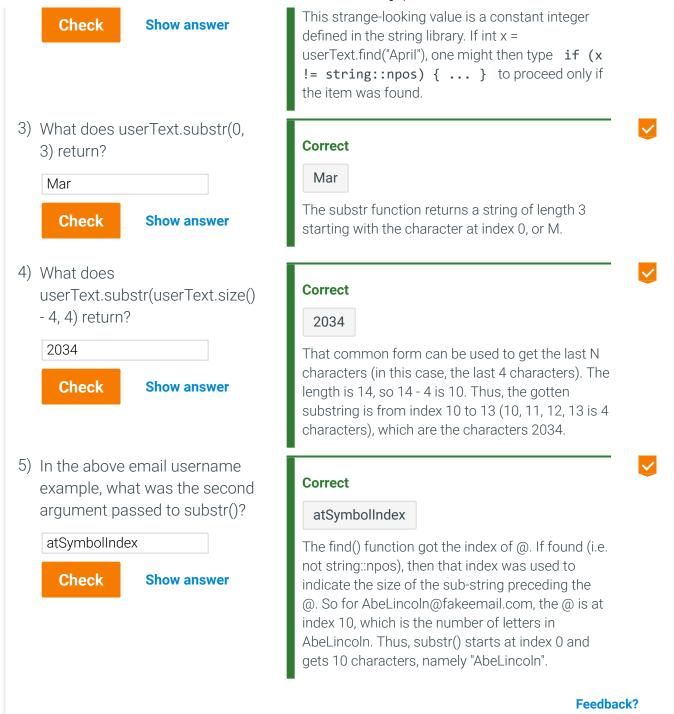
```
#include <iostream>
#include <string>
using namespace std;
int main() {
   string emailText;
   int atSymbolIndex;
   string emailUsername;
   cout << "Enter email address: ";</pre>
   cin >> emailText;
   atSymbolIndex = emailText.find('@');
   if (atSymbolIndex == string::npos) {
      cout << "Address is missing @" << endl;</pre>
   else {
      emailUsername = emailText.substr(0,
atSymbolIndex);
      cout << "Username: " << emailUsername <<</pre>
end1;
   return 0;
}
```

Enter email address:
AbeLincoln@fakeemail.com
Username: AbeLincoln
...
Enter email address: swimming_is_fun
Address is missing @

Feedback?

PARTICIPATION 3.15.1: find() and substr(). **ACTIVITY** userText is "March 17, 2034". Do not type quotes in answers. 1) What does userText.find(',') Correct return? 8 8 ", is the 9th character, so the index is 8. Check **Show answer** 2) What does userText.find("April") **Correct** return? string::npos string::npos

3.15. More string operations



Combining / Replacing

The string library has more functions for modifying strings.

Table 3.15.2: String modify functions, invoked as myString.push_back(c). Each increases/decreases string's length appropriately.

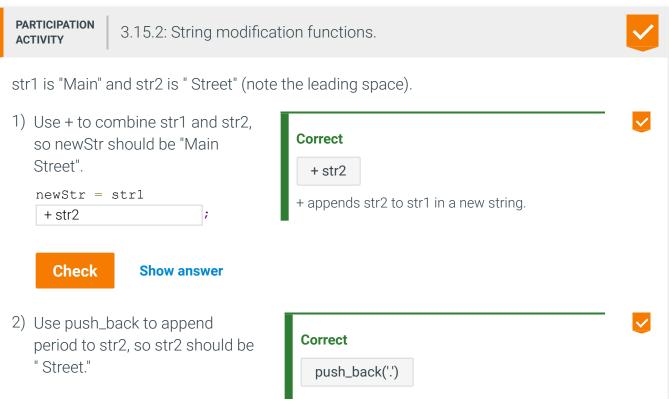
```
push_back()
               push_back(c)
                                // userText is "Hello"
                                 userText.push_back('?'); // Now "Hello?"
               appends
                                 userText.size();
                                                        // Returns 6
               character c
               to the end of
               a string.
insert()
               insert(indx,
                                 // userText is "Goodbye"
                                 userText.insert(0, "Well "); // Now "Well Goodbye"
               subStr)
               Inserts string
                                 // userText is "Goodbye"
                                 userText.insert(4, "---"); // Now "Good---bye"
               subStr
               starting at
               index indx.
replace()
               replace(indx,
                                 // userText is "You have many gifts"
                                 userText.replace(9, 4, "a plethora of");
               num, subStr)
                                 // Now "You have a plethora of gifts"
               replaces
               characters at
               indices indx
               to indx+num-
               1 with a copy
               of subStr.
str1 + str2
               Returns a
                                 // userText is "A B"
                                 myString = userText + " C D";
               new string
                                 // myString is "A B C D"
                                 myString = myString + '!';
               that is a copy
                                 // myString now "A B C D!"
               of str1 with
                                 myString = myString + userText;
                                 // myString now "A B C D!A B"
               str2
               appended.
               If one of str1,
               str2 is a
               string, the
               other may be
               a character
               (or character
               array).
```

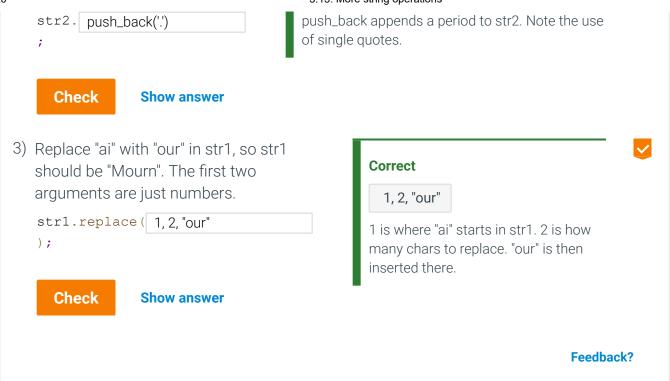
Feedback?

Figure 3.15.2: String modify example: Greeting.

```
#include <iostream>
                                                                   Enter name: Julia
#include <string>
                                                                   Hello Julia.
using namespace std;
                                                                   Hello Mr/Ms Julia.
                                                                   Hello Mr/Ms Julia.
int main() {
   string userName;
   string greetingText;
   int itemIndex;
                                                                   Enter name: Darn Rabbit
                                                                   Hello Darn Rabbit.
   itemIndex = 0;
                                                                   Hello Mr/Ms Darn Rabbit.
                                                                   Hello Mr/Ms @#$ Rabbit.
   cout << "Enter name: ";</pre>
   getline(cin, userName);
   // Combine strings using +
   greetingText = "Hello " + userName;
   // Append a period (could have used +)
   greetingText.push_back('.'); // '' not ""
   cout << greetingText << endl;</pre>
   // Insert Mr/Ms before user's name
   greetingText.insert(6, "Mr/Ms ");
   cout << greetingText << endl;</pre>
   // Replace occurrence of "Darn" by "@$#"
   if (greetingText.find("Darn") != string::npos) { // Found
      itemIndex = static_cast<int>(greetingText.find("Darn"));
      greetingText.replace(itemIndex, 4, "@#$");
   cout << greetingText << endl;</pre>
   return 0;
```

Feedback?





Exploring further:

11

Numerous additional functions exist for strings.

• C++ string library

```
CHALLENGE
             3.15.1: Combining strings.
ACTIVITY
Assign secretID with firstName, a space, and lastName. Ex: If firstName is Barry and
lastName is Allen, then output is:
Barry Allen
    1 #include <iostream>
      #include <string>
    3 using namespace std;
    5 int main() {
         string secretID;
    6
         string firstName;
    8
         string lastName;
    9
         cin >> firstName;
   10
```

cin >> lastName;

```
13
14
        /* Your solution goes here */
        secretID = firstName + " " + lastName;
  15
  16
        cout << secretID << endl;</pre>
  17
        return 0;
  18 }
           All tests passed
  Run

✓ Testing: "Barry", "Allen"

           Your output
                            Barry Allen

✓ Testing: "Steve", "Austin"

           Your output
                            Steve Austin

✓ Testing: "Selina", "Kyle"

           Your output
                            Selina Kyle
                                                                                  Feedback?
```

CHALLENGE ACTIVITY

3.15.2: Name song.



Modify songVerse to play "The Name Game" (see OxfordDictionaries.com), by replacing "(Name)" with userName but without the first letter.

Ex: If userName = "Katie" and songVerse = "Banana-fana fo-f(Name)!", the program prints:

Banana-fana fo-fatie!

Ex: If userName = "Katie" and songVerse = "Fee fi mo-m(Name)", the program prints:

Fee fi mo-matie

Note: You may assume songVerse will always contain the substring "(Name)".

```
3 using namespace std;
4
5 int main() {
```

```
string userName;
    6
         string songVerse;
    8
    9
         getline(cin, userName);
         userName = userName.substr(1, userName.size() - 1); // Remove first character
   10
   11
   12
         getline(cin, songVerse);
   13
         // Modify songVerse to replace (Name) with userName without first character
   14
   15
         /* Your solution goes here */
   16
   17
            songVerse.replace(songVerse.find('('), 6, userName);
   18
   19
         cout << songVerse << endl;</pre>
   20
   21
         return 0;
   22
   23 }
  Run
           All tests passed
✓ Testing Katie and Banana-fana fo-f(Name)!
            Your output
                            Banana-fana fo-fatie!

✓ Testing Walter and Banana-fana fo-f(Name)!

            Your output
                            Banana-fana fo-falter!

✓ Testing Katie and Fee fi mo-m(Name)

            Your output
                            Fee fi mo-matie
                                                                                 Feedback?
CHALLENGE
             3.15.3: Using find().
ACTIVITY
Print "Censored" if userInput contains the word "darn", else print userInput. End with
newline. Ex: If userInput is "That darn cat.", then output is:
Censored
Ex: If userInput is "Dang, that was scary!", then output is:
Dang, that was scary!
```

Note: If the submitted code has an out-of-range access, the system will stop running the code after a few seconds, and report "Program end never reached." The system doesn't print the test case that caused the reported message.

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
5 int main() {
      string userInput;
6
7
      getline(cin, userInput);
8
9
      /* Your solution goes here */
10
11
      //cout << substr(userInput.find('d'), 4);</pre>
      if(userInput.substr(userInput.find('d'), 4) == "darn" || userInput.substr(userInput.f
12
13
          cout << "Censored";</pre>
14
      }else{
15
          cout << userInput;</pre>
16
17
18
19
      cout << endl;</pre>
       notunn A
20
21
```

Run

✓ Testing for: That darn cat.

Your output Censored

X Test aborted

```
Exited with return code -6 (SIGIOT).
terminate called after throwing an instance of 'std::out_of_range
  what(): basic_string::substr: __pos (which is 1844674407370955
```

Feedback?