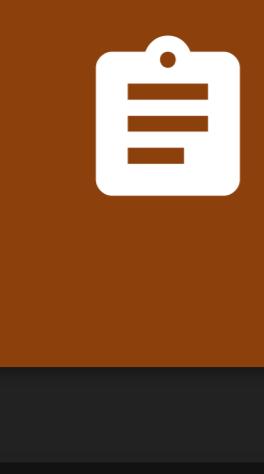


[↑4.5 More for loop examples](#)

Students:  
Section 4.6 is a part of 2 assignments: [CSC108 CH04.1-4.6 C4A](#)

Please browse to this assignment through BlackboardLearn so zyBooks knows where to send your activity. [Learn more](#)

Includes: CA  
Due: 03/11/2025, 11:59 PM EDT

## 4.6 Loops and strings

### Iterating through a string with a for loop

A programmer commonly iterates through a string, examining each character. The following example counts the number of letters in a string, not counting digits, symbols, etc.

Figure 4.6.1: Iterating through a string: Counting letters.

```
#include <iostream>
#include <string>
#include <cctype>
using namespace std;

int main() {
    string inputWord;
    int numLetters;
    unsigned int i;

    cout << "Enter a word: ";
    cin >> inputWord;

    numLetters = 0;
    for (i = 0; i < inputWord.size(); ++i) {
        if (isalpha(inputWord.at(i))) {
            numLetters += 1;
        }
    }

    cout << "Number of letters: " << numLetters << endl;
}

return 0;
}
```

Enter a word: Hey!!  
Number of letters: 3  
...

Enter a word: 123abc...xyz  
Number of letters: 6

[Feedback?](#)

PARTICIPATION ACTIVITY

4.6.1: Iterating through a string.

- 1) To visit every character in a string, a for loop should iterate over indices \_\_\_\_.
  - 0 to size
  - 0 to size-1
  - 1 to size
- 2) If a for loop iterates through a string s using variable i, the loop body can access the current character as:  
s.at(\_\_\_\_)
  - i-1
  - i
  - i+1

[Feedback?](#)

### Iterating until done with a while loop

A programmer commonly wishes to iterate through a string until something is done. The example below replaces all occurrences of "U.S.A." with "USA". Because the number of iterations is not known beforehand, a while loop is used. The string functions find() and replace() are used to identify each instance of the "U.S.A." and replace each instance with "USA", respectively. Both functions are described in detail elsewhere.

Figure 4.6.2: Iterating until done: Replacing all occurrences of a word.

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    string userText;
    int usaIndex;

    cout << "Enter text: ";
    getline(cin, userText);

    // At least one occurrence exists
    while (userText.find("U.S.A.") != string::npos) {
        // Get index of first instance
        usaIndex = userText.find("U.S.A.");

        // U.S.A. is 6 long
        userText.replace(usaIndex, 6, "USA");
    }

    cout << "New text: " << userText << endl;

    return 0;
}
```

Enter text: The U.S.A. is big. Are you from the U.S.A.?  
New text: The USA is big. Are you from the USA?  
...

Enter text: USA U.S.A. U.S.A.U.S.A. Bye  
New text: USA USA USAUSA Bye

[Feedback?](#)

PARTICIPATION ACTIVITY

4.6.2: Replacing until done.

Consider the example above.

- 1) The loop is entered as long as an occurrence of "U.S.A." \_\_\_\_.
  - is found
  - is not found
- 2) The number of iterations is known before entering the loop.
  - True
  - False
- 3) find() is called within the loop body.
  - True
  - False
- 4) replace() is called with the index of the first occurrence of "U.S.A.", \_\_\_\_ , and "USA" as the arguments.
  - 1
  - 6

[Feedback?](#)

CHALLENGE ACTIVITY

4.6.1: Password requirements.

620890.5010016.qx3zqy7

[Start](#)

A website requires that passwords only contain numbers. For each character in password that is *not* a number, assign the character with '9' (the number nine).

[Click for example](#) ▾

Note: isdigit() returns true if a character is a digit, and false otherwise. Ex: isdigit('8') returns true. isdigit('a') returns false.

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 int main() {
6     string password;
7     unsigned int i;
8
9     getline(cin, password);
10
11    /* Your code goes here */
12
13    cout << "Adjusted password: " << password << endl;
14
15    return 0;
16 }
```

1

2

3

[Check](#)

[Next level](#)

[Feedback?](#)

How was this section? [Provide section feedback](#)

Activity summary for assignment: [CSC108 CH04.1-4.6 C4A](#)

Due: 03/11/2025, 11:59 PM EDT

5 / 36 points

Please browse to this assignment through BlackboardLearn so zyBooks knows where to send your activity. [Learn more](#)

[Completion details](#) ▾

[↓4.7 Nested loops](#)