Students:
Section 6.16 is a part of 1 assignment: **CSC108 CH06.10-6.29 P6B**

Please browse to this assignment through BlackboardLearn so zyBooks knows where to send your activity. Learn more

Includes: PA
Due: 05/06/2025, 11:59 PM EDT

6.16 Char library functions: cctype

C++ provides common functions for working with characters, presented in the cctype library. The first c indicates the library is a C language standard library, and ctype is short for "character type". To use those functions, the programmer adds the following at the top of a file:
`#include <cctype>`

Commonly-used cctype functions are summarized below; a complete reference is found at the cctype reference page.

Character checking functions

The following functions check whether a character is of a given category, returning either false (0) or true (non-zero).

Table 6.16.1: Functions that check whether a character is of a given category.

The examples below assume the following string declaration.

```
char myString[30] = "Hey9! Go";
```

isalpha(c) -- Returns true if c is alphabetic: a-z or A-Z.	`isalpha('A');` // Returns true `isalpha(myString[0]);` // Returns true because 'H' is alphabetic `isalpha(myString[3]);` // Returns false because '9' is not alphabetic
isdigit(c) -- Returns true if c is a numeric digit: 0-9.	`isdigit(myString[3]);` // Returns true because '9' is numeric `isdigit(myString[4]);` // Returns false because '!' is not numeric
isalnum(c) -- Returns true if c is alphabetic or a numeric digit. Thus, returns true if either isalpha or isdigit would return true.	`isalnum('A');` // Returns true `isalnum(myString[3]);` // Returns true because '9' is numeric
isspace(c) -- Returns true if character c is a whitespace.	`isspace(myString[5]);` // Returns true because that character is a space ' '. `isspace(myString[0]);` // Returns false because 'H' is not whitespace.
islower(c) -- Returns true if character c is a lowercase letter a-z.	`islower(myString[0]);` // Returns false because 'H' is not lowercase. `islower(myString[1]);` // Returns true because 'e' is lowercase. `islower(myString[3]);` // Returns false because '9' is not a lowercase letter.
isupper(c) -- Returns true if character c is an uppercase letter A-Z.	`isupper(myString[0]);` // Returns true because 'H' is uppercase. `isupper(myString[1]);` // Returns false because 'e' is not uppercase. `isupper(myString[3]);` // Returns false because '9' is not an uppercase letter.
isblank(c) -- Returns true if character c is a blank character. Blank characters include spaces and tabs.	`isblank(myString[5]);` // Returns true because that character is a space ' '. `isblank(myString[0]);` // Returns false because 'H' is not blank.
isxdigit(c) -- Returns true if c is a hexadecimal digit: 0-9, a-f, A-F.	`isxdigit(myString[3]);` // Returns true because '9' is a hexadecimal digit. `isxdigit(myString[1]);` // Returns true because 'e' is a hexadecimal digit. `isxdigit(myString[6]);` // Returns false because 'G' is not a hexadecimal digit.
ispunct(c) -- Returns true if c is a punctuation character. Punctuation characters include: !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~	`ispunct(myString[4]);` // Returns true because '!' is a punctuation character. `ispunct(myString[6]);` // Returns false because 'G' is not a punctuation character.
isprint(c) -- Returns true if c is a printable character. Printable characters include alphanumeric, punctuation, and space characters.	`isprint(myString[0]);` // Returns true because 'H' is a alphabetic. `isprint(myString[4]);` // Returns true because '!' is punctuation. `isprint(myString[5]);` // Returns true because that character is a space ' '. `isprint('\0');` // Returns false because the null character is not printable
iscntrl(c) -- Returns true if c is a control character. Control characters are all characters that are not printable.	`iscntrl(myString[0]);` // Returns false because 'H' is a not a control character `iscntrl(myString[5]);` // Returns false because space is a not a control character `iscntrl('\0');` // Returns true because the null character is a control character

Feedback?

Character conversion functions

The following functions return a character representing a converted version of the input character.

Table 6.16.2: Functions that convert a character to upper or lower case.

The examples below assume the following string declaration.

```
char myString[30] = "Hey9! Go";
```

toupper(c) -- If c is a lowercase alphabetic character (a-z), returns the uppercase version (A-Z). If c is not a lowercase alphabetic character, just returns c.	`letter = toupper(myString[0]);` // Returns 'H' (no change) `letter = toupper(myString[1]);` // Returns 'E' ('e' converted to 'E') `letter = toupper(myString[3]);` // Returns '9' (no change) `letter = toupper(myString[5]);` // Returns ' ' (no change)
tolower(c) -- If c is an uppercase alphabetic character (A-Z), returns the lowercase version (a-z). If c is not an uppercase alphabetic character, just returns c.	`letter = tolower(myString[0]);` // Returns 'h' ('H' converted to 'h') `letter = tolower(myString[1]);` // Returns 'e' (no change) `letter = tolower(myString[3]);` // Returns '9' (no change) `letter = tolower(myString[5]);` // Returns ' ' (no change)

Feedback?

The following example illustrates some of the cctype functions.

Figure 6.16.1: Use of some functions in cctype.

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

int main() {
   const int MAX_LEN = 30;       // Max string length
   char userStr[MAX_LEN];        // User defined string
   int i;

   // Prompt user to enter string
   cout << "Enter string (<"
   << MAX_LEN << " chars): ";
   cin >> userStr;

   cout << "Original: " << userStr << endl;

   cout << "isalpha:  ";
   for (i = 0; userStr[i] != '\0'; ++i) {
      if (isalpha(userStr[i])) {
         cout << "Y";
      }
      else {
         cout << "N";
      }
   }
   cout << endl;

   cout << "isdigit:  ";
   for (i = 0; userStr[i] != '\0'; ++i) {
      if (isdigit(userStr[i])) {
         cout << "Y";
      }
      else {
         cout << "N";
      }
   }
   cout << endl;

   cout << "isupper:  ";
   for (i = 0; userStr[i] != '\0'; ++i) {
      if (isupper(userStr[i])) {
         cout << "Y";
      }
      else {
         cout << "N";
      }
   }
   cout << endl;

   for (i = 0; userStr[i] != '\0'; ++i) {
      userStr[i] = toupper(userStr[i]);
   }
   cout << "After toupper: " << userStr << endl;

   return 0;
}
```

```
Enter string (<30 chars): ABC123!def
Original: ABC123!def
isalpha:  YYYNNNNNYYY
isdigit:  NNNYYYNNNNN
isupper:  YYYNNNNNNNN
After toupper: ABC123!DEF
```

Feedback?

To compare two strings without paying attention to case, one technique is to first convert (a copy of) each string to lowercase (using a loop, discussed elsewhere) and then comparing.

PARTICIPATION ACTIVITY | 6.16.1: Character type functions.

Enter the value to which each function evaluates using 1 for true, 0 for false for boolean functions.
Assume str is "Hi 321".

1) isalpha(str[0])

Check Show answer

2) isdigit(str[4])

Check Show answer

3) isalnum(str[2])

Check Show answer

4) isspace(str[2])

Check Show answer

5) islower(str[6])

Check Show answer

6) tolower(str[0])

Check Show answer

7) tolower(str[1])

Check Show answer

Feedback?

How was this section? | Provide section feedback

Activity summary for assignment: CSC108 CH06.10-6.29 P6B
Due: 05/06/2025, 11:59 PM EDT

0 / 91 points

Please browse to this assignment through BlackboardLearn so zyBooks knows where to send your activity. Learn more

Completion details