

Hands-on Activity 4.4

Characters and Strings

Course Code: CPE 007

Program: Computer Engineering

Course Title: Programming Logic and Design

Date Performed: 9/25/2025

Section: CPE11S1

Date Submitted: 9/25/2025

Name(s): Juan Paulo C. Lara

Instructor: Engr. Jimlord M. Quejado

6. Output

1.1 islower

CODE

```
1 #include <iostream>
2 #include <cctype>
3 #include <cstring>
4 using namespace std;
5 int main() {
6     char ch;
7
8     cout << "According to islower:\n\n";
9
10    for (int i=0; i<4; i++) {
11        cout << "Input a letter: ";
12        cin >> ch;
13
14        if (islower(ch)) {
15            cout << ch << " is a lowercase letter\n";
16        } else {
17            cout << ch << " is not a lowercase letter\n";
18        }
19    }
20    return 0;
21 }
```

OUTPUT

```
According to islower:

Input a letter: p
p is a lowercase letter
Input a letter: P
P is not a lowercase letter
Input a letter: 5
5 is not a lowercase letter
Input a letter: !
! is not a lowercase letter

-----
Process exited after 9.705 seconds with return value 0
Press any key to continue . . .
```

ANALYSIS

During execution of this code, line 6 first declares ch as char for the following commands that will use it. Line 8 prints a label stating the function this program will use, which is islower(ch). A for loop at line 10 looks for the 4 inputs that will be given by the user according to how the sample output looks like. Line 11 and 12 tells the user to input a lower case letter.

Then, an if statement at line 14 checks whether the input letter is a lowercase. The next line is its then statement outputting that the input letter is a lowercase letter. An else statement at line 16 and 17 prints out stating that the inputted letter is not a lowercase letter.

1.2 isupper

CODE

```
1 #include <iostream>
2 #include <cctype>
3 #include <cstring>
4 using namespace std;
5 int main() {
6     char ch;
7
8     cout << "According to isupper:\n\n";
9
10    for (int i=0; i<4; i++) {
11        cout << "Input a letter: ";
12        cin >> ch;
13
14        if (isupper(ch)) {
15            cout << ch << " is a uppercase letter\n";
16        } else {
17            cout << ch << " is not an uppercase letter\n";
18        }
19    }
20    return 0;
21 }
22 |
```

OUTPUT

```
According to isupper:

Input a letter: D
D is a uppercase letter
Input a letter: d
d is not an uppercase letter
Input a letter: 8
8 is not an uppercase letter
Input a letter: &
& is not an uppercase letter

-----
Process exited after 16.78 seconds with return value 0
Press any key to continue . . .
```

ANALYSIS

This code is similar to the previous one, which asks for islower. But in this instance, it asks for isupper or upper case. During execution, line 6 declares ch as char for the next commands to use. Line 8 prints out the label for the use of this program, which is to determine whether the input letter is uppercase or not. A for loop at line 10 scans for the 4 inputs of the next two lines 11-12. The two lines wait for user input of an uppercase letter. Line 14 and 15 contains the if function and expression (isupper(ch)) to output if the input letter is uppercase. An else statement at line 16 and its command at line 17, print if the input is not an uppercase letter.

1.3 Conversion

CODE

```
1 #include <iostream>
2 #include <cctype>
3 #include <cstring>
4 using namespace std;
5 int main() {
6     char ch;
7
8     cout << "CONVERSION\n\n";
9
10    for (int i=0; i<4; i++) {
11        cout << "Input a letter: ";
12        cin >> ch;
13
14        if (isupper(ch)) {
15            cout << ch << " converted to lowercase is " << (char)tolower(ch) << endl;
16        } else if (islower(ch)) {
17            cout << ch << " converted to uppercase is " << (char)toupper(ch) << endl;
18        } else {
19            cout << ch << " converted to uppercase is " << ch << endl;
20        }
21    }
22
23 }
```

OUTPUT

```
CONVERSION

Input a letter: u
u converted to uppercase is U
Input a letter: 7
7 converted to uppercase is 7
Input a letter: $
$ converted to uppercase is $
Input a letter: L
L converted to lowercase is l

-----
Process exited after 10.38 seconds with return value 0
Press any key to continue . . .
```

ANALYSIS

When running this code, line 6 declares ch as char for the next blocks of commands to use. Line 8 labels the program's function to convert upper case letters to lower case ones or vice versa. A for loop at line 10 scans for the 4 user inputs of the next two lines. Line 11 and 12 within the loop waits for user input of a letter in each iteration. Three following if else statements still inside the for loop starting at line 14 checking the given input of ch. Line 15 converts the given input through (char)tolower(ch), which means to convert upper case ch to lower case ch. At line 16, an else if statement similar to the previous one also looks for ch, but this time it's for the lower case. Its next line converts the given lowercase to an uppercase equivalent, similar to the previous one at line 15. An else statement at line 18 catches for an input aside from uppercase or lowercase and pastes it as is.

7. Supplementary Activity

#2
CODE

```
1 #include <iostream>
2 #include <cctype>
3 #include <cstring>
4 using namespace std;
5 int main() {
6     char ch;
7     cout << "Input a character: ";
8     ch = cin.get();
9
10    switch (ch) {
11        case ' ':
12            cout << "Received: [SPACE]";
13            break;
14        case '\t':
15            cout << "Received: [TAB]";
16            break;
17        case '\n':
18            cout << "Received: [ENTER]";
19            break;
20        default:
21            cout << "Received: " << ch;
22    }
23
24    cout << "\nCHARACTER TEST\n";
25    cout << "isalnum: " << (isalnum(ch) ? "Yes" : "No") << endl;
26    cout << "isalpha: " << (isalpha(ch) ? "Yes" : "No") << endl;
27    cout << "isdigit: " << (isdigit(ch) ? "Yes" : "No") << endl;
28    cout << "islower: " << (islower(ch) ? "Yes" : "No") << endl;
29    cout << "isupper: " << (isupper(ch) ? "Yes" : "No") << endl;
30    cout << "isblank: " << (isblank(ch) ? "Yes" : "No") << endl;
31    cout << "isspace: " << (isspace(ch) ? "Yes" : "No") << endl;
32    cout << "iscntrl: " << (iscntrl(ch) ? "Yes" : "No") << endl;
33    cout << "isprint: " << (isprint(ch) ? "Yes" : "No") << endl;
34    cout << "ispunct: " << (ispunct(ch) ? "Yes" : "No") << endl;
35    cout << "isxdigit: " << (isxdigit(ch) ? "Yes" : "No") << endl;
36    cout << "tolower: " << (char)tolower(ch) << endl;
37    cout << "toupper: " << (char)toupper(ch) << endl;
38
39    return 0;
40 }
```

OUTPUT

```
Input a character:  
Received: [SPACE]  
CHARACTER TEST  
isalnum: No  
isalpha: No  
isdigit: No  
islower: No  
isupper: No  
isblank: Yes  
isspace: Yes  
iscntrl: No  
isprint: Yes  
ispunct: No  
isxdigit: No  
tolower:  
toupper:
```

```
-----  
Process exited after 1.611 seconds with return value 0  
Press any key to continue . . .
```

```
Input a character:  
Received: [ENTER]  
CHARACTER TEST  
isalnum: No  
isalpha: No  
isdigit: No  
islower: No  
isupper: No  
isblank: No  
isspace: Yes  
iscntrl: Yes  
isprint: No  
ispunct: No  
isxdigit: No  
tolower:  
toupper:
```

```
-----  
Process exited after 1.156 seconds with return value 0  
Press any key to continue . . .
```

```
Input a character:
```

```
Received: [TAB]
```

```
CHARACTER TEST
```

```
isalnum: No
```

```
isalpha: No
```

```
isdigit: No
```

```
islower: No
```

```
isupper: No
```

```
isblank: Yes
```

```
isspace: Yes
```

```
iscntrl: Yes
```

```
isprint: No
```

```
ispunct: No
```

```
isxdigit: No
```

```
tolower:
```

```
toupper:
```

```
-----
```

```
Process exited after 0.9518 seconds with return value 0
```

```
Press any key to continue . . .
```

```
Input a character: Z
```

```
Received: Z
```

```
CHARACTER TEST
```

```
isalnum: Yes
```

```
isalpha: Yes
```

```
isdigit: No
```

```
islower: No
```

```
isupper: Yes
```

```
isblank: No
```

```
isspace: No
```

```
iscntrl: No
```

```
isprint: Yes
```

```
ispunct: No
```

```
isxdigit: No
```

```
tolower: z
```

```
toupper: Z
```

```
-----
```

```
Process exited after 0.8137 seconds with return value 0
```

```
Press any key to continue . . . -
```

```
Input a character: z
```

```
Received: z
```

```
CHARACTER TEST
```

```
isalnum: Yes
```

```
isalpha: Yes
```

```
isdigit: No
```

```
islower: Yes
```

```
isupper: No
```

```
isblank: No
```

```
isspace: No
```

```
iscntrl: No
```

```
isprint: Yes
```

```
ispunct: No
```

```
isxdigit: No
```

```
tolower: z
```

```
toupper: Z
```

```
Process exited after 0.6709 seconds with return value 0
```

```
Press any key to continue . . .
```

```
Input a character: 0xFFFFF
Received: 0
CHARACTER TEST
isalnum: Yes
isalpha: No
isdigit: Yes
islower: No
isupper: No
isblank: No
isspace: No
iscntrl: No
isprint: Yes
ispunct: No
isxdigit: Yes
tolower: 0
toupper: 0

-----
Process exited after 3.203 seconds with return value 0
Press any key to continue . . .
```

```
Input a character: @
Received: @
CHARACTER TEST
isalnum: No
isalpha: No
isdigit: No
islower: No
isupper: No
isblank: No
isspace: No
iscntrl: No
isprint: Yes
ispunct: Yes
isxdigit: No
tolower: @
toupper: @

-----
Process exited after 1.57 seconds with return value 0
Press any key to continue . . .
```

ANALYSIS

During execution of this code, line 6 declares ch for the next commands to use. Line 7 and 8 waits for user input. Compared to previous codes, instead of a `cin >> ch`, it uses `ch = cin.get()`. This is mainly to allow for more inputs than the traditional `cin` method as it allows for characters such as space, tab, and enter. A switch statement at line 10 checks for the value of ch with the following cases at line 11, 14, and 17. Line 11 checks for space and outputs the corresponding input and this applies for line 14 and 17 with their respective inputs. A default statement at line 20 receives and prints the given input ch. Line 24 labels the function of this program as a test for characters based on the table given by the activity. Starting from line 25 to 35, each function looks for the input given by the user and determines whether the input corresponds to the function or not with a yes or no answer. Line 36 and 37 like the previous codes, simply convert the given input to their uppercase or lowercase through the functions.

#3

CODE

```
1 #include <iostream>
2 #include <cctype>
3 #include <cstring>
4 using namespace std;
5 int main() {
6     string a, b, c, d;
7     int sum;
8
9     cout << "Input 4 numbers\n";
10    cout << "1. : ";
11    cin >> a;
12    cout << "2. : ";
13    cin >> b;
14    cout << "3. : ";
15    cin >> c;
16    cout << "4. : ";
17    cin >> d;
18
19    int int1 = stoi(a);
20    int int2 = stoi(b);
21    int int3 = stoi(c);
22    int int4 = stoi(d);
23
24    sum = int1 + int2 + int3 + int4;
25    cout << "Sum: " << sum << endl;
26
27    return 0;
28 }
```

OUTPUT

```
Input 4 numbers
1. : 25
2. : 50
3. : 67
4. : 33
Sum: 175

-----
Process exited after 7.657 seconds with return value 0
Press any key to continue . . .
```

ANALYSIS

When running this program, line 6 first declares a,b,c,d as strings for the inputs given by the user. The next line declares the sum of all the four numbers that will be computed later. A simple input and output mechanism starts from line 9 to 17. Line 9 tells the user to input the desired numbers. Line 10, 12, 14, and 16 outputs the number of times the numbers have been entered. Line 11, 13, 15 and 17 waits for user input one number at a time. From line 19 to 22, a conversion from string to int involves the use of the stoi command, which converts the given string a to an int value int1 that the sum computation can understand. At line 24, all the converted values are added to the sum variable, then the next line outputs the sum.

8. Conclusion

Concluding this activity, I have learned even more of the ways the C++ language can store data and other forms of information through the use of string and its libraries cstring and cctype. This topic also introduced me to the many functions and handlers that check for various user inputs of letters, symbols, characters, and print lines. With the ASCII

codes discussed during the preliminary period of this course, I realized that these inputs are more or less addresses that the computer uses to read and write data from one variable to the next. The string command allows for more flexibility in data flow and control, and also allows for conversions to different formats based on the supplementary activities. Doing this activity also encouraged me to do more research on data types, libraries, and other commands that can be useful in managing data in code.