

## Hands-on Activity 4.4:

### Characters and Strings

Course Code: CPE007

Program: Computer Engineering

Course Title: Programming Logic and Design

Date Performed: 9/25/2025

Section: CPE11S1

Date Submitted: 9/25/2025

Name(s): Ramirez, Angel Mae C.

Instructor: Engr. Jimlord M. Quejado

## 6. Output

1.  
CODE:

```
[*] Untitled1.cpp
1  #include <iostream>
2  #include <cctype>
3
4  using namespace std;
5
6  int main() {
7
8      cout << "According to islower:\n";
9      char chars1[] = {'p', 'P', '5', '!'};
10     int size1 = sizeof(chars1) / sizeof(chars1[0]);
11     for (int i = 0; i < size1; i++) {
12         char c = chars1[i];
13         if (islower(c))
14             cout << c << " is a lowercase letter\n";
15         else
16             cout << c << " is not a lowercase letter\n";
17     }
18
19     cout << "\nAccording to isupper:\n";
20     char chars2[] = {'D', 'd', '8', '&'};
21     int size2 = sizeof(chars2) / sizeof(chars2[0]);
22     for (int i = 0; i < size2; i++) {
23         char c = chars2[i];
24         if (isupper(c))
25             cout << c << " is an uppercase letter\n";
26         else
27             cout << c << " is not an uppercase letter\n";
28     }
29
30
31     cout << "\n";
32     cout << "u converted to uppercase is " << (char)toupper('u') << endl;
33     cout << "7 converted to uppercase is " << (char)toupper('7') << endl;
34     cout << "$ converted to uppercase is " << (char)toupper('$') << endl;
35     cout << "L converted to lowercase is " << (char)tolower('L') << endl;
36
37     return 0;
38 }
```

Compile Log   Debug   Find Results

Sel: 0   Lines: 39   Length: 1167   Insert   Done parsing in 0.031 s

## OUTPUT:

```
C:\Users\Zarina\Documents\L x + v - □
According to islower:
p is a lowercase letter
P is not a lowercase letter
5 is not a lowercase letter
! is not a lowercase letter

According to isupper:
D is an uppercase letter
d is not an uppercase letter
8 is not an uppercase letter
& is not an uppercase letter

u converted to uppercase is U
7 converted to uppercase is 7
$ converted to uppercase is $
L converted to lowercase is l

-----
Process exited after 0.1961 seconds with return value 0
Press any key to continue . . . |
```

## ANALYSIS:

In this output activity the code is suppose to check if a character is uppercase or lowercase and also convert characters using functions like `islower()`, `isupper()`, `toupper()` and `tolower()` from the `cctype` library. At first I wrote the code using a rangebased for loop cause I saw that in other examples, but when I try to run it it didn't work. I got confused because the syntax seemed right, but it turns out that range-based loops don't always work well with normal arrays. I didn't know that the compiler can't always figure out the size of the array when using that kind of loop. I had to look it up and change it to a normal for loop using the array size and index to fix it. That part took me a while because I wasn't sure how to calculate the array size correctly at first, but then I remembered about using ``sizeof()`` for that. I also didn't know at first that ``toupper()`` and ``tolower()`` return an `int`, not a `char`, so when I print the result it was showing weird values. So I added a cast to `char` when printing it out and that made it work how I expected. The code now checks if a character is lowercase or uppercase and tells the user, and also shows what the conversion would look like. This activity was kinda hard at first but I learned a lot about how `c++` handles characters and also got better at reading error messages and figuring out what's wrong.

## 7. Supplementary Activity

2. Write a program that inputs a character from the keyboard and tests the character with each of the functions in the character handling library. (Refer to the first Table above)

CODE:

Untitled1.cpp

```
1  #include <iostream>
2  #include <cctype>
3
4  using namespace std;
5
6  int main() {
7      char ch;
8
9      cout << "Enter a character: ";
10     cin >> ch;
11
12     cout << "\nCharacter analysis:\n";
13
14     if (isalnum(ch))
15         cout << ch << " is alphanumeric.\n";
16     else
17         cout << ch << " is not alphanumeric.\n";
18
19     if (isalpha(ch))
20         cout << ch << " is an alphabet.\n";
21     else
22         cout << ch << " is not an alphabet.\n";
23
24     if (isdigit(ch))
25         cout << ch << " is a digit.\n";
26     else
27         cout << ch << " is not a digit.\n";
28
29     if (islower(ch))
30         cout << ch << " is a lowercase letter.\n";
31     else
32         cout << ch << " is not a lowercase letter.\n";
33
34     if (isupper(ch))
35         cout << ch << " is an uppercase letter.\n";
36     else
37         cout << ch << " is not an uppercase letter.\n";
38
```

Untitled1.cpp

```
39     if (isblank(ch))
40         cout << ch << " is a blank character (space or tab).\n";
41     else
42         cout << ch << " is not a blank character.\n";
43
44     if (isspace(ch))
45         cout << ch << " is a whitespace character.\n";
46     else
47         cout << ch << " is not a whitespace character.\n";
48
49     if (iscntrl(ch))
50         cout << ch << " is a control character.\n";
51     else
52         cout << ch << " is not a control character.\n";
53
54     if (isprint(ch))
55         cout << ch << " is a printable character.\n";
56     else
57         cout << ch << " is not a printable character.\n";
58
59     if (ispunct(ch))
60         cout << ch << " is a punctuation character.\n";
61     else
62         cout << ch << " is not a punctuation character.\n";
63
64     if (isxdigit(ch))
65         cout << ch << " is a hexadecimal digit.\n";
66     else
67         cout << ch << " is not a hexadecimal digit.\n";
68
69     cout << "\nConversions:\n";
70     cout << "toupper(" << ch << ") = " << (char)toupper(ch) << endl;
71     cout << "tolower(" << ch << ") = " << (char)tolower(ch) << endl;
72
73     return 0;
74 }
75
```

OUTPUT:

```
C:\Users\Zarina\Documents\l  X  +  v
Enter a character: A

Character analysis:
A is alphanumeric.
A is an alphabet.
A is not a digit.
A is not a lowercase letter.
A is an uppercase letter.
A is not a blank character.
A is not a whitespace character.
A is not a control character.
A is a printable character.
A is not a punctuation character.
A is a hexadecimal digit.

Conversions:
toupper(A) = A
tolower(A) = a

-----
Process exited after 5.693 seconds with return value 0
Press any key to continue . . . |
```

```
C:\Users\Zarina\Documents\l  X  +  v
Enter a character: M

Character analysis:
M is alphanumeric.
M is an alphabet.
M is not a digit.
M is not a lowercase letter.
M is an uppercase letter.
M is not a blank character.
M is not a whitespace character.
M is not a control character.
M is a printable character.
M is not a punctuation character.
M is not a hexadecimal digit.

Conversions:
toupper(M) = M
tolower(M) = m

-----
Process exited after 7.379 seconds with return value 0
Press any key to continue . . .
```

```
C:\Users\Zarina\Documents\L  X + v
Enter a character: R

Character analysis:
R is alphanumeric.
R is an alphabet.
R is not a digit.
R is not a lowercase letter.
R is an uppercase letter.
R is not a blank character.
R is not a whitespace character.
R is not a control character.
R is a printable character.
R is not a punctuation character.
R is not a hexadecimal digit.

Conversions:
toupper(R) = R
tolower(R) = r

-----
Process exited after 5.447 seconds with return value 0
Press any key to continue . . . |
```

## ANALYSIS:

In this activity, I created a program that lets the user type any character, and then it checks that character using all the functions from the character handling library in C++. I tested it using the letters a, m, and r, because those are my initials hehe and I just wanted to see what kind of output they would give. I used functions like `isalpha()`, `isdigit()`, `isupper()`, `islower()`, `isblank()`, and more. At first, it was kinda confusing because there are so many different functions and they all sound the same. I mixed up `isspace()` and `isblank()` and didn't understand what the difference was, but after trying both I saw that they test different types of space. One problem I ran into was with the conversion functions `toupper()` and `tolower()`. I thought they would just print the uppercase or lowercase version directly, but it turns out they return an int so you have to cast it to a char. That made me think the code was broken at first but then I figured it out. I also didn't expect that `isxdigit()` would say that some letters are hexadecimal digits. I thought it was just numbers. Overall, this code helped me learn a lot about how to test characters properly and understand what C++ thinks about letters, spaces, digits, and other symbols. I think I get it now more than before.

3. Write a program that inputs 4 strings that represent integers, converts the strings to integers, sums the values and prints the total of the 4 values.

CODE:

```
Untitled1.cpp
1  #include <iostream>
2  #include <cstdlib>
3
4  using namespace std;
5
6  int main() {
7      char str1[20], str2[20], str3[20], str4[20];
8
9      cout << "Enter 4 numbers (as strings):\n";
10
11     cout << "Number 1: ";
12     cin >> str1;
13
14     cout << "Number 2: ";
15     cin >> str2;
16
17     cout << "Number 3: ";
18     cin >> str3;
19
20     cout << "Number 4: ";
21     cin >> str4;
22
23     // Convert to integers
24     int num1 = atoi(str1);
25     int num2 = atoi(str2);
26     int num3 = atoi(str3);
27     int num4 = atoi(str4);
28
29     int sum = num1 + num2 + num3 + num4;
30
31     cout << "\nThe sum of the numbers is: " << sum << endl;
32
33     return 0;
34 }
35
```

OUTPUT:

```
C:\Users\Zarina\Documents\l × + ∨
Enter 4 numbers (as strings):
Number 1: 56
Number 2: 70
Number 3: 47
Number 4: 80

The sum of the numbers is: 253

-----
Process exited after 41.69 seconds with return value 0
Press any key to continue . . . |
```

```
C:\Users\Zarina\Documents\l x + v
Enter 4 numbers (as strings):
Number 1: AB
Number 2: B
Number 3: H
Number 4: TH

The sum of the numbers is: 0

-----
Process exited after 17.7 seconds with return value 0
Press any key to continue . . .
```

## ANALYSIS:

For this code, the task was to write a program that takes 4 string inputs that represent numbers, then converts them into integers and prints the total sum. I started by thinking maybe I could just use int for the inputs directly, but then I realized the point was to use strings first, then convert them. So I used char str1[20], str2, str3, and str4 to hold the values, cause each input might be up to 20 characters and that's usually enough. Then I used the atoi() function from the <cstdlib> library to convert each string to an integer. I didn't really use atoi() before so I had to look it up, and also I forgot to include the right library at first so it gave me an error. But once I added #include <cstdlib> it started working fine.

Once I fixed that, I did int num1 = atoi(str1); and did the same for the other 3 strings. Then I just added them together with int sum = num1 + num2 + num3 + num4; and used cout to print the total. I tested the program by entering 56, 70, 47, and 80 as the inputs (all strings), and the program gave me the result 253, which was the correct answer. That's when I knew the logic was working. One thing that confused me is that atoi() doesn't give an error if you type letters, it just returns 0. So if the user types something like "abc" or "12ab", the program won't complain but the result will be wrong. I didn't add validation yet but I understand now why it's important.

## 8. Conclusion

Through all of these activities, I learned a lot about how characters and strings really work in c++. Before this, I honestly didn't know there was so many different functions just to check things like if a letter is uppercase or if a space is printable. Using the ctype library and trying out functions like isalpha(), isdigit(), isupper(), islower(), and others helped me see how the computer actually understands letters and symbols. I also learned how you can take a single character and find out all these different things about it just by using simple functions.

Another thing I learned is how to change a character's case, like making lowercase into uppercase using toupper() and vice versa with tolower(). I didn't realize that those functions return an integer so I got confused at first why the result wasn't printing right, but once I figured out I needed to cast it back to char it worked. I also really liked the part where I had to change spaces into new lines and test things like control characters and punctuation. It made me think about how every single character has a value and meaning to the computer.

Lastly, the part about string conversions using atoi() was actually very useful. I never used that function before and now I understand how you can take string input that looks like a number and actually do math with it. It helped me understand how input validation is important too because if someone types letters instead of numbers, the result can go wrong. Overall, these activities gave me better confidence in handling strings and characters in C++. It was kind of confusing at first, but by testing everything myself and fixing errors, I actually understand it now.

