

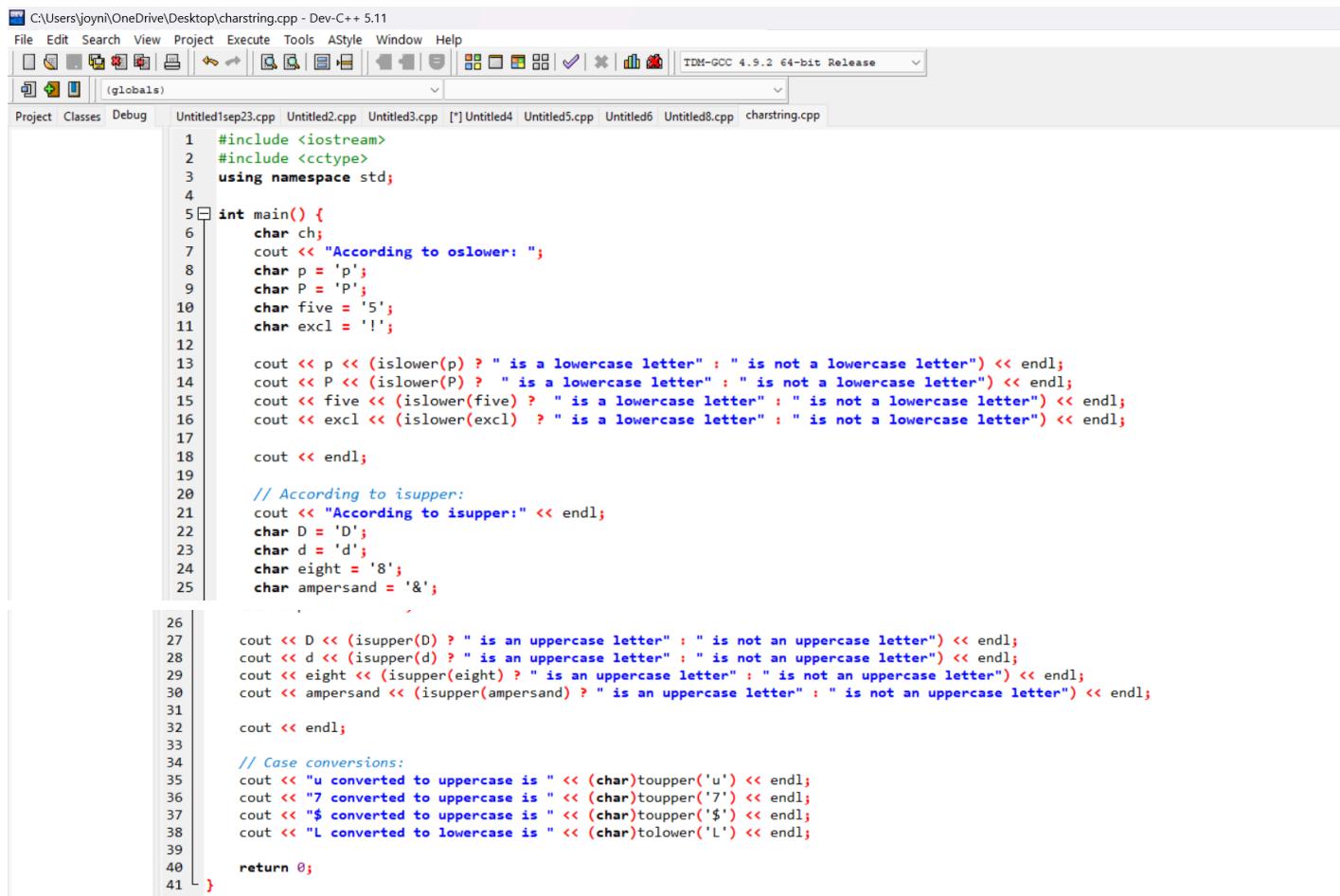
## Activity No. n4.4

### Replace with Title

<b>Course Code:</b> CPE007	<b>Program:</b> Computer Engineering
<b>Course Title:</b> Programming Logic and Design	<b>Date Performed:</b> 09/23/25
<b>Section:</b> CPE11S1	<b>Date Submitted:</b> 09/25/25
<b>Name(s):</b> Cenndy M.Nieles	<b>Instructor:</b> Engr. Jimlord M. Quejado

### 6. Output

INPUT:



The screenshot shows the Dev-C++ IDE interface with the file "charstring.cpp" open. The code implements various character classification functions using `<iostream>` and `<cctype>`. It includes sections for lowercase and uppercase letter checks, as well as case conversions. The code uses `cout` statements to output results to the console.

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

int main() {
    char ch;
    cout << "According to oslower: ";
    char p = 'p';
    char P = 'P';
    char five = '5';
    char excl = '!';
    cout << p << (islower(p) ? " is a lowercase letter" : " is not a lowercase letter") << endl;
    cout << P << (islower(P) ? " is a lowercase letter" : " is not a lowercase letter") << endl;
    cout << five << (islower(five) ? " is a lowercase letter" : " is not a lowercase letter") << endl;
    cout << excl << (islower(excl) ? " is a lowercase letter" : " is not a lowercase letter") << endl;
    cout << endl;

    // According to isupper:
    cout << "According to isupper:" << endl;
    char D = 'D';
    char d = 'd';
    char eight = '8';
    char ampersand = '&';

    cout << D << (isupper(D) ? " is an uppercase letter" : " is not an uppercase letter") << endl;
    cout << d << (isupper(d) ? " is an uppercase letter" : " is not an uppercase letter") << endl;
    cout << eight << (isupper(eight) ? " is an uppercase letter" : " is not an uppercase letter") << endl;
    cout << ampersand << (isupper(ampersand) ? " is an uppercase letter" : " is not an uppercase letter") << endl;
    cout << endl;

    // Case conversions:
    cout << "u converted to uppercase is " << (char) toupper('u') << endl;
    cout << "7 converted to uppercase is " << (char) toupper('7') << endl;
    cout << "$ converted to uppercase is " << (char) toupper('$') << endl;
    cout << "L converted to lowercase is " << (char) tolower('L') << endl;

    return 0;
}
```

OUTPUT:

```
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 0.1491 seconds with return value 0
Press any key to continue . . . |
```

Compiler: TDM-GCC 4.9.2 64-bit Release  
Abort Compilation  
shorten compiler paths - Warnings: 0 - Output Filename: C:\Users\joyni\OneDrive\Desktop\charstring.exe

**EXPLANATION:** Four char variables are declared: p, P, five, and excl. They are initialized with a special character, a digit, an uppercase letter, and a lowercase letter, respectively. Each character is printed using cout, along with a message indicating whether or not islower() returns true (a lowercase letter) or false (a non-lowercase letter). Do you employ the ternary operator condition? value\_if\_true: value\_if\_false to select the relevant message according to the islower() result. supper tests You declare four char variables (D, d, eight, and ampersand) and use cout and the ternary operator to print whether isupper() returns true or false for each character. This is similar to the islower tests. Conversions of Cases. To convert the letter "u" to uppercase, use the toupper() function. Toupper() will return 'U' because 'u' is a lowercase letter. Prior to printing, the outcome is cast to char. You apply toupper() to the character '7'. Toupper() will return '7' unaltered because it is not a lowercase letter. You apply toupper() to the character ". " Toupper() will return '\$' unaltered because " is not a lowercase letter. You use tolower() to convert the character 'L' to lowercase. Tolower() will return 'l' since 'L' is an uppercase letter. Prior to printing, the outcome is cast to char.

## 7. Supplementary Activity

INPUT:

C:\Users\joyni\OneDrive\Desktop\Untitled10.cpp - Dev-C++ 5.11

```
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release

Project Classes Debug Untitled1.cpp Untitled2.cpp [*]Untitled3.cpp Untitled4.cpp Untitled5.cpp Untitled6.cpp Untitled7.cpp [*]Untitled8.cpp [*]Untitled9.cpp Untitled10.cpp

1 #include <iostream>
2 #include <cctype>
3 #include <algorithm> // for sort()
4 using namespace std;
5
6 int main() {
7     char ch;
8     cout << "Enter a character: ";
9     cin >> ch;
10
11    if (isalnum(ch)) {
12        cout << "" << ch << "' is alphanumeric (letter or digit)." << endl;
13    } else {
14        cout << "" << ch << "' is not alphanumeric." << endl;
15    }
16
17    if (isalpha(ch)) {
18        cout << "" << ch << "' is an alphabet letter." << endl;
19    } else {
20        cout << "" << ch << "' is not an alphabet letter." << endl;
21    }
22}
```

C:\Users\joyni\OneDrive\Desktop\Untitled10.cpp - Dev-C++ 5.11

```
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release

Project Classes Debug Untitled1.cpp Untitled2.cpp [*]Untitled3.cpp Untitled4.cpp Untitled5.cpp Untitled6.cpp Untitled7.cpp [*]Untitled8.cpp [*]Untitled9.cpp Untitled10.cpp

19 } else {
20     cout << "" << ch << "' is not an alphabet letter." << endl;
21 }
22
23 if (isblank(ch)) {
24     cout << "" << ch << "' is a blank character (space or tab)." << endl;
25 } else {
26     cout << "" << ch << "' is not blank." << endl;
27 }
28
29 if (iscntrl(ch)) {
30     cout << "" << ch << "' is a control character." << endl;
31 } else {
32     cout << "" << ch << "' is not a control character." << endl;
33 }
34
35 if (isdigit(ch)) {
36     cout << "" << ch << "' is a digit." << endl;
37 } else {
38     cout << "" << ch << "' is not a digit." << endl;
39 }
40
41 if (islower(ch)) {
42     cout << "" << ch << "' is lowercase." << endl;
43 } else {
44     cout << "" << ch << "' is not lowercase." << endl;
45 }
46
47 if (isprint(ch)) {
48     cout << "" << ch << "' is printable." << endl;
49 } else {
50     cout << "" << ch << "' is not printable." << endl;
51 }
52
53 if (ispunct(ch)) {
54     cout << "" << ch << "' is punctuation." << endl;
55 } else {
56     cout << "" << ch << "' is not punctuation." << endl;
57 }
58
59 if (isspace(ch)) {
```

```

60     cout << "" << ch << "' is whitespace." << endl;
61 } else {
62     cout << "" << ch << "' is not whitespace." << endl;
63 }
64
65 if (isupper(ch)) {
66     cout << "" << ch << "' is uppercase." << endl;
67 } else {
68     cout << "" << ch << "' is not uppercase." << endl;
69 }
70
71 if (isxdigit(ch)) {
72     cout << "" << ch << "' is a hexadecimal digit (0-9, a-f, A-F)." << endl;
73 } else {
74     cout << "" << ch << "' is not a hexadecimal digit." << endl;
75 }
76
77 if (isalpha(ch)) {
78     cout << "Lowercase version: " << (char)tolower(ch)
79     << ", Uppercase version: " << (char)toupper(ch) << endl;
80 } else {
81     cout << "Case conversion does not apply (not a letter)." << endl;
82 }
83
// Sorting array of characters
84 char arr[] = {'z', 'a', 'c', 'x', 'b'};
85 int n = sizeof(arr) / sizeof(arr[0]);
86
87 sort(arr, arr + n);
88 cout << "\nSorted characters: ";
89 for (int i = 0; i < n; i++) {
90     cout << arr[i] << " ";
91 }
92 cout << endl;
93
94 return 0;
95
96
97

```

## OUTPUT:

```

Enter a character: a
'a' is alphanumeric (letter or digit).
'a' is an alphabet letter.
'a' is not blank.
'a' is not a control character.
'a' is not a digit.
'a' is lowercase.
'a' is printable.
'a' is not punctuation.
'a' is not whitespace.
'a' is not uppercase.
'a' is a hexadecimal digit (0-9, a-f, A-F).
Lowercase version: a, Uppercase version: A

Sorted characters: a b c x z

Process exited after 4.978 seconds with return value 0
Press any key to continue . . .

```

source  
n  
aths  
- Output Size: 1.8547477722168 MiB  
- Compilation Time: 0.52s

## EXPLANATION:

Utilizing the library. After receiving a single character from the user, it performs a number of checks to identify the character type. It determines whether the character is a digit, a lowercase letter, a printable character, a punctuation symbol, a whitespace character, an uppercase letter, a hexadecimal digit, an alphabetic character, a blank space or tab, a control character, or an alphanumeric (letter or number). If the character is a letter, the program displays both its lowercase and uppercase versions after classifying it. Lastly, the program sorts a small array of characters (`{'z', 'a', 'c', 'x', 'b'}`) and prints them in alphabetical order to show how to use the `sort()` function from .

To put it briefly, this application teaches you how to use C++'s character classification, case conversion, and basic sorting functions.

```
main.cpp
1 #include <iostream>
2 #include <string>
3 #include <cstdlib>
4 using namespace std;
5
6 int main() {
7     string str1, str2, str3, str4;
8     int sum = 0;
9
10    cout << "Enter the first integer string: ";
11    cin >> str1;
12    cout << "Enter the second integer string: ";
13    cin >> str2;
14    cout << "Enter the third integer string: ";
15    cin >> str3;
16    cout << "Enter the fourth integer string: ";
17    cin >> str4;
18
19    sum = stoi(str1) + stoi(str2) + stoi(str3) + stoi(str4);
20
21    cout << "The total sum of the four values is: " << sum << endl;
22
23    return 0;
24 }
25 }
```

OUTPUT:

Output

Clear

```
Enter the first integer string: 1
Enter the second integer string: 2
Enter the third integer string: 3
Enter the fourth integer string: 4
The total sum of the four values is: 10

==> Code Execution Successful ==>
```

OUTPUT:

C++ program that takes four numbers as input from the user, but instead of reading them as integers directly, it first stores them as strings. It then uses the `stoi()` function to convert each string into an integer. After converting all four strings, the program adds them together and stores the result in a variable called `sum`. Finally, it displays the total sum to the user on the screen. In short, your program demonstrates how to take string inputs, convert them into integers, perform arithmetic operations, and output the result.

## 8. Conclusion

Programmers can determine whether a character in C++ is a letter, digit, space, punctuation, or another specific type by using a range of functions from the library. Characters can be easily categorized and processed using these functions without requiring a manual comparison to ASCII values. Additionally, `toupper()` and `tolower()` allow characters to be changed from uppercase to lowercase, which is useful for case-insensitive comparisons or text formatting. Sorting characters enables us to place them in a particular order, typically determined by their ASCII values. Sorting tasks like alphabetical sorting is made simple by the `sort()` function from `<algorithm>`, which allows us to easily arrange characters in ascending order. Character sorting and classification work together to offer strong tools for effectively managing and arranging textual data in C++.