Activity No. 4.4	
Characters and Strings	
Course Code: CPE007	**Program:** Computer Engineering
Course Title: Programming Logic and Design	**Date Performed:9/23/25**
Section:CPE11S1	**Date Submitted:9/23/25**
Name(s): Will Stuart D. Ponce Jr.	**Instructor:** Engr. Jimlord M. Quejado

6. 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

```cpp
#include <iostream>
#include <cctype>

int main() {
    std::cout << "According to islower:" << std::endl;
    std::cout << "p is " << (islower('p') ? "" : "not ") << "a lowercase letter" << std::endl;
    std::cout << "P is " << (islower('P') ? "" : "not ") << "a lowercase letter" << std::endl;
    std::cout << "5 is " << (islower('5') ? "" : "not ") << "a lowercase letter" << std::endl;
    std::cout << "! is " << (islower('!') ? "" : "not ") << "a lowercase letter" << std::endl;

    return 0;
}
```

Explanation:

Through the usage of the C++ language, this program imparts a very understandable example of the character-handling functions from the <cctype> library. First of all the program includes <iostream> and <cctype> headers, <iostream> for outputting to the console and <cctype> for the functions themselves. Next, the code interrogates the cases of several characters ('p', 'P', '5', etc.) by employing the slower() and isupper() functions and to print the results it uses a compact ternary operator where the word "not" is conditionally added to the output string. Moreover, the code demonstrates character conversion with the help of toupper() and tolower() by changing characters from one case to another and non-alphabetic characters are left unchanged. The (char) cast is used on the results of toupper() and tolower() so that the output is a character and not the integer ASCII value.

```
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

--------------------------------
Process exited after 0.01435 seconds with return value 0
Press any key to continue . . .
```

2.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

```cpp
#include <iostream>
#include <cctype>

int main() {
    std::cout << "\nAccording to isupper:" << std::endl;
    std::cout << "D is " << (isupper('D') ? "" : "not ") << "an uppercase letter" << std::endl;
    std::cout << "d is " << (isupper('d') ? "" : "not ") << "an uppercase letter" << std::endl;
    std::cout << "8 is " << (isupper('8') ? "" : "not ") << "an uppercase letter" << std::endl;
    std::cout << "& is " << (isupper('&') ? "" : "not ") << "an uppercase letter" << std::endl;
    std::cout << std::endl;

    return 0;
}
```

Explanation:

The C++ code you have chosen is aimed at the demonstration of the basic functionalities of the character handling library. Initially, it includes the <iostream> header which is used for the output to the console and <cctype> that is a library containing the functions required for the character manipulation. The program then goes on to test all the different characters by using islower() and isupper() functions to check their case, as well as, a ternary operator to convert the output in a user-friendly format based on the boolean value. After the classification tests, it presents the change of case by giving the examples of the toupper() and tolower() functions on a set of characters. Later, the results of all these function invocations are printed by the standard output, thus, the direct and easy-to-understand example of how each function operates with different characters is provided.

```
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


--------------------------------
Process exited after 0.1061 seconds with return value 0
Press any key to continue . . .
```

3.u converted to uppercase is U

7 converted to uppercase is 7

$ converted to uppercase is $

L converted to lowercase is l

[*] Untitled1.cpp

```cpp
#include <iostream>
#include <cctype>

int main() {
    std::cout << "'u' converted to uppercase is " << (char)toupper('u') << std::endl;
    std::cout << "'7' converted to uppercase is " << (char)toupper('7') << std::endl;
    std::cout << "'$' converted to uppercase is " << (char)toupper('$') << std::endl;
    std::cout << "'L' converted to lowercase is " << (char)tolower('L') << std::endl;

    return 0;
}
```

Explanation:

This code snippet written in C++ really shows in a clearer way how a case switching is done by the help of the <cctype> library. Various characters were selected, and the output was produced by the use of std::cout, calling the toupper() and tolower() functions one after another. The example with the toupper() function is the conversion of the lower case letter 'u' into its upper case equivalent 'U'. Non-alphabetic characters like '7' or '$' are left unchanged, and the function tolower() only changes the uppercase letter 'L' into the lowercase one 'l'. Casting of the (char) type is a very important operation in this code and it is absolutely necessary to do it as toupper() and tolower() actually give back an integer, which is the ASCII value of the character; the cast makes sure that the printed result is a character and not a number.

```
'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 0.1205 seconds with return value 0
Press any key to continue . . .
```

7. Supplementary Activity

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)

```cpp
#include <iostream>
#include <cctype>
void print_test(const char* func_name, bool result) {
    std::cout << func_name << ": " << (result ? "Yes" : "No") << std::endl;
}
int main() {
    char input_char;
    std::cout << "Enter a single character to test: ";
    std::cin >> input_char;
    std::cout << "\n--- Analysis for character '" << input_char << "' ---" << std::endl;
    print_test("Is Alphanumeric? (isalnum)", isalnum(input_char));
    print_test("Is Alphabetic?   (isalpha)", isalpha(input_char));
    print_test("Is Lowercase?    (islower)", islower(input_char));
    print_test("Is Uppercase?    (isupper)", isupper(input_char));
    print_test("Is a Digit?      (isdigit)", isdigit(input_char));
    print_test("Is Hex Digit?    (isxdigit)", isxdigit(input_char));
    print_test("Is Control Char? (iscntrl)", iscntrl(input_char));
    print_test("Is Printable?    (isprint)", isprint(input_char));
    print_test("Has Graphics?    (isgraph)", isgraph(input_char));
    print_test("Is Whitespace?   (isspace)", isspace(input_char));
    print_test("Is Punctuation?  (ispunct)", ispunct(input_char));
    std::cout << "-----------------------------------" << std::endl;
    std::cout << "Converted to uppercase: " << (char)toupper(input_char) << std::endl;
    std::cout << "Converted to lowercase: " << (char)tolower(input_char) << std::endl;

    return 0;
}
```

Explanation:

This piece of code in C++ is an interactive single-character analysis program that takes input from the user via the keyboard. For console input/output operations, it utilizes the <iostream> library, and for character classification and conversion, it employs the <cctype> library, which is a complete set of functionsThe program after the prompt acquires the user input, reads the character, then hands it off to a user-defined auxiliary function print_test, which effectively arranges and presents the outcome of the various tests like isalnum, isalpha, and isdigit for the given characterThe program sequentially invokes print_test for each one of the eleven character classification functions, thus, it provides an exhaustive input analysis. In the end, it features the case conversion functions, toupper and tolower, just to be sure that the shown results are characters not their ASCII codes, it explicitly converts their output back to char.

```
Enter a single character to test: A

--- Analysis for character 'A' ---
Is Alphanumeric? (isalnum): Yes
Is Alphabetic?   (isalpha): Yes
Is Lowercase?    (islower): No
Is Uppercase?    (isupper): Yes
Is a Digit?      (isdigit): No
Is Hex Digit?    (isxdigit): Yes
Is Control Char? (iscntrl): No
Is Printable?    (isprint): Yes
Has Graphics?    (isgraph): Yes
Is Whitespace?   (isspace): No
Is Punctuation?  (ispunct): No
-----------------------------------
Converted to uppercase: A
Converted to lowercase: a

--------------------------------
Process exited after 11.62 seconds with return value 0
Press any key to continue . . .
```

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.

```cpp
#include <iostream>
#include <string>
#include <cstdlib>

int main() {

    std::string s1, s2, s3, s4;


    std::cout << "Enter 4 integer values (as text), separated by spaces: ";
    std::cin >> s1 >> s2 >> s3 >> s4;


    int num1 = atoi(s1.c_str());
    int num2 = atoi(s2.c_str());
    int num3 = atoi(s3.c_str());
    int num4 = atoi(s4.c_str());


    int sum = num1 + num2 + num3 + num4;


    std::cout << "The values you entered are: " << num1 << ", " << num2 << ", " << num3 << ", and " << num4 << "." << std::endl;
    std::cout << "Their total sum is: " << sum << std::endl;

    return 0;
    }
```

Explanation:

The purpose of this C++ program is to get four numbers in text form, convert them into integers, and add them up. It is a program that uses the <iostream>, <string>, and <cstdlib> libraries to handle input/output from the console, deal with strings, and help the conversion from string to integer, respectively. The program asks the user to input four values separated by spaces, which are then saved in four different std::string variables.The main part of the program is the atoi function call that converts each of the four inputs from strings to integers. After the conversion, the program calculates and outputs the sum of the four integers to the console so that the user can see the numbers and their sum.

```
Enter 4 integer values (as text), separated by spaces: s1
s2
s3
s4
The values you entered are: 0, 0, 0, and 0.
Their total sum is: 0

--------------------------------
Process exited after 9.965 seconds with return value 0
Press any key to continue . . .
```

8.Conclusion:All the C++ programs that you have provided collectively are very good to use as one whole course for the introduction of the characters and strings manipulation fundamental concepts in C++. Firstly, the programs effectively demonstrate the use of the character-handling library, <cctype>, by referring to the functions like islower, isupper, tolower, and toupper for the characters classification and conversion. This kind of an investigation equips with the basic understanding of C++ concept on how standard data types are handled. Besides, the extra programs expand the users' interaction with the tools; the new user-driven character input is analyzed fully, and the output defines all the properties of the character, thus, the user considered to be the very helpful function of the library.The last program, string_sum.cpp, is vital as it shifts the focus from individual characters to strings which stand for numbers. By showing how the user input in plain text can be changed into a format that is suitable for the calculation, the program string_sum.cpp successfully captures the attention.