Activity No. 4.4	
Hands-on Activity 4.4: Characters and Strings	
Course Code: CPE007	Program: Computer Engineering
Course Title: Programming Logic and Design	Date Performed: September 24, 2025
Section: CPE11S1	Date Submitted: September 25, 2025
Name(s): Tobias, Lawrence C.	Instructor: Engr. Jimlord M. Quejado
6. Output	

1.

Na para bang.cpp | hindi ka gaganyan.cpp | Pahghudd.cpp | idk.cpp

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

int main () {
    char ch[5] = "Dd8&";

    for (int i = 0; i < 4; i++){
        if (isupper(ch[i])){
            cout << ch[i] << " Is a Uppercase letter\n";
        }
        else{
            cout << ch[i] << " Is a Lowercase letter\n";
        }
    }

    return 0;
}
```

```
C:\Dev-Cpp\Na para bang.exe
D Is a Uppercase letter
d Is a Lowercase letter
8 Is a Lowercase letter
& Is a Lowercase letter

--------------------------------
Process exited after 2.08 seconds with return value 0
Press any key to continue . . .
```

- This code checks if the characters are uppercase or not. The program goes through the characters "D", "d", "8", and "&". On line 5 I made a character array with those four elements, and on line 7 I made a for loop that starts at index 0 and runs until it checks all the characters. On lines 8–12 I used an If..else statement that first checks if the character is uppercase and then checks if it is lowercase. Because of this, the program correctly shows "D" as uppercase, "d" as lowercase, and does not wrongly say that "8" and "&" are lowercase.

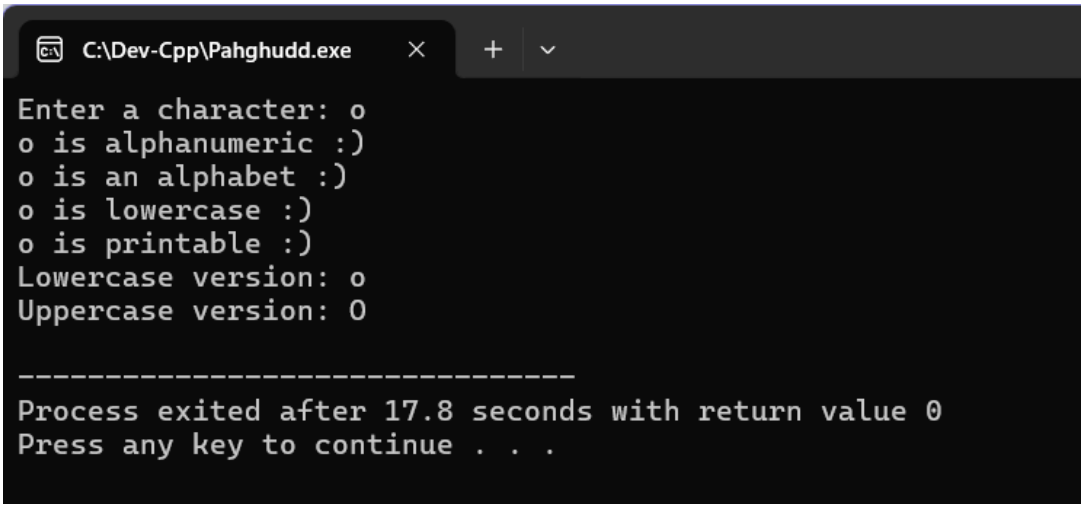
7. Supplementary Activity

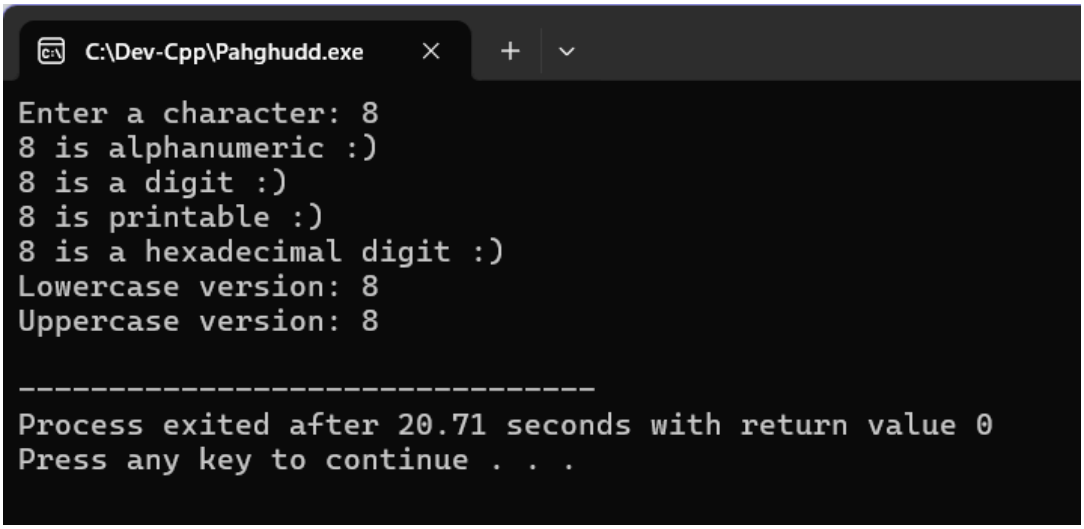
2.

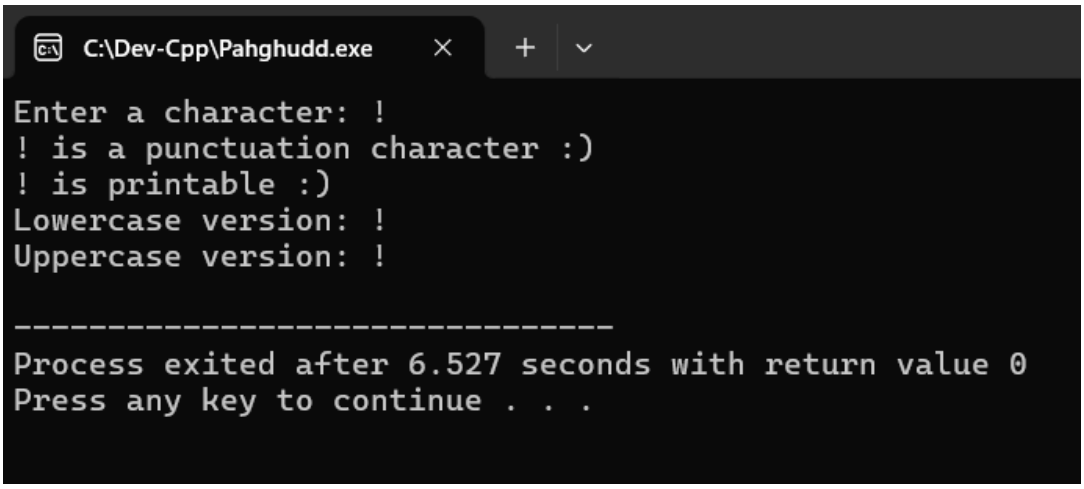
Na para bang.cpp | hindi ka gaganyan.cpp | Pahghudd.cpp | idk.cpp

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

int main() {
    char ch;
    cout << "Enter a character: ";
    cin >> ch;

    if (isalnum(ch))
        cout << ch << " is alphanumeric :) " << endl;
    if (isalpha(ch))
        cout << ch << " is an alphabet :) " << endl;
    if (isdigit(ch))
        cout << ch << " is a digit :) " << endl;
    if (islower(ch))
        cout << ch << " is lowercase :) " << endl;
    if (isupper(ch))
        cout << ch << " is uppercase :) " << endl;
    if (isspace(ch))
        cout << "You entered a whitespace character :) " << endl;
    if (isblank(ch))
        cout << "You entered a blank space character :) " << endl;
    if (ispunct(ch))
        cout << ch << " is a punctuation character :) " << endl;
    if (isprint(ch))
        cout << ch << " is printable :) " << endl;
    if (iscntrl(ch))
        cout << "You entered a control character :) " << endl;
    if (isxdigit(ch))
        cout << ch << " is a hexadecimal digit :) " << endl;

    cout << "Lowercase version: " << (char)tolower(ch) << endl;
    cout << "Uppercase version: " << (char)toupper(ch) << endl;

    return 0;
}
```

```
C:\Dev-Cpp\Pahghudd.exe
Enter a character: T
T is alphanumeric :)
T is an alphabet :)
T is uppercase :)
T is printable :)
Lowercase version: t
Uppercase version: T

--------------------------------
Process exited after 3.965 seconds with return value 0
Press any key to continue . . .
```







- This code checks different kinds of characters that the user inputs. The program asks the user to enter a character and then tests it in many ways. On line 5 I declared a character variable ch to store the input, and on lines 6–7 the program asks the user to type a character and saves it in ch. On lines 9–29 the program uses many If statements to check the character, like if it is alphanumeric, an alphabet, a digit, lowercase, uppercase, a space, a blank, a punctuation, printable, a control character, or a hexadecimal digit. On lines 31–32 the program also shows the lowercase version and the uppercase version of the character. Because of this, the program can tell the user exactly what kind of character they entered and also show its converted forms.

3.

Na para bang.cpp | hindi ka gaganyan.cpp | Pahghudd.cpp | idk.cpp

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

int stringToInt(string s) {
    int num = 0;
    for (int i = 0; i < s.length(); i++) {
        if (isdigit(s[i])) {
            num = num * 10 + (s[i] - '0');
        }
    }
    return num;
}

int main() {
    string s1, s2, s3, s4;

    cout << "Enter 4 numbers (as strings): ";
    cin >> s1 >> s2 >> s3 >> s4;

    int n1 = stringToInt(s1);
    int n2 = stringToInt(s2);
    int n3 = stringToInt(s3);
    int n4 = stringToInt(s4);

    int total = n1 + n2 + n3 + n4;

    cout << "The total is: " << total << endl;

    return 0;
}
```

```
C:\Dev-Cpp\idk.exe
Enter 4 numbers (as strings): 0
1
1
3
The total is: 5

--------------------------------
```

- This code changes string inputs into integers and then adds them together. On lines 6–13 I made a function called stringToInt that converts a string into an integer by starting with a value of 0, looping through each character, and if the character is a digit, multiplying the current number by 10 and adding the digit. After the loop, the function returns the number. On line 16 I declared four string variables s1, s2, s3, and s4, and on lines 18–19 the program asks the user to enter four numbers as strings and saves them in those variables. On lines 21–24 each string is converted into an integer using the stringToInt function and stored in n1, n2, n3, and n4. On line 26 I declared a variable total that stores the sum of all four integers, and on line 28 the program prints the total sum. Because of this, the program can take numbers written as strings, change them into integers, and calculate their total correctly.

8. Conclusion

- In this lesson I learned how character functions work and how they are used in a program. I was able to check and identify letters, numbers, symbols, and even spaces. I also practiced explaining codes step by step, like declaring variables, using loops, and applying If…else statements to make decisions. The activities helped me understand more about functions, such as checking if a character is uppercase, lowercase, a digit, or even changing strings into integers. It was good practice for me even though I made some mistakes and felt confused at times. In the end, I know I still need to practice more, but I now have a better understanding of how to use these functions in my codes.