**String Operations (Length, Substring and Find)**

In C++, the **std::string** class provides several member functions and operators for performing common string operations, such as finding the length of a string, extracting substrings, and searching for substrings within a string. Here are a few examples of common string operations in C++:

1. Finding the length of a string: To find the length of a string, you can use the **length** member function of the **std::string** class. The **length** function returns the number of characters in the string, not including the null terminator.

For example:

C++

#include <iostream>

#include <string>

int main()

{

std::string str = "hello";

std::cout << "The length of the string is " << str.length() << " characters." << std::endl;

return 0;

}  
**Output**

The length of the string is 5 characters.

      2.Extracting a substring: To extract a substring from a string, you can use the **substr** member function of the **std::string** class. The **substr** function takes two                arguments: the starting position of the substring, and the length of the substring.

For example:

C++

#include <iostream>

#include <string>

int main()

{

std::string str = "hello world";

std::string sub = str.substr(6, 5); // extract the substring starting at position 6 with a length of 5 characters

std::cout << "The substring is: " << sub << std::endl;

return 0;

}

**Output**

The substring is: world

Searching for a substring: To search for a substring within a string, you can use the find member function of the std::string class. The find function searches for a given substring within a string and returns the position of the first occurrence of the substring, or `std::string::n

The **find** function takes two arguments: the substring to search for, and the starting position of the search. By default, the search starts at the beginning of the string, but you can specify a different starting position if you want to search only a portion of the string.

Here is an example of how to use the **find** function to search for a substring within a string starting at a specific position:

C++

#include <iostream>

#include <string>

int main()

{

std::string str = "hello world";

std::size\_t pos = str.find("l", 3); // search for the first occurrence of 'l' starting at position 3

if (pos != std::string::npos) // check if the substring was found

{

std::cout << "The substring was found at position " << pos << std::endl;

}

else

{

std::cout << "The substring was not found." << std::endl;

}

return 0;

}

**Output**

The substring was found at position 3

Note that the **find** function is case-sensitive, so it will only find substrings that match the case of the search string. If you want to perform a case-insensitive search, you can use the **find** function in combination with the **tolower** or **toupper** functions from the **cctype** library.

C++

#include <cctype> // include the cctype library for tolower and toupper

#include <iostream>

#include <string>

int main()

{

std::string str = "Hello World";

std::string search\_string = "world";

for (char& c : search\_string) // convert the search string to lowercase

{

c = std::tolower(c);

}

std::size\_t pos = str.find(search\_string); // search for the lowercase version of the search string

if (pos != std::string::npos) // check if the substring was found

{

std::cout << "The substring was found at position " << pos << std::endl;

}

else

{

std::cout << "The substring was not found." << std::endl;

}

return 0;

}

**Output**

The substring was not found.