

C++ Programming

- C++ Flow Control ▶
- C++ Functions ▶
- C++ Arrays & String ▼
 - C++ Arrays
 - Multidimensional Arrays
 - C++ Function and Array
 - » **C++ String**
- C++ Structures ▶
- C++ Object & Class ▶
- C++ Pointers ▶
- C++ Inheritance ▶



Related Examples

Title
C++ Program to Find the Frequency of Characters in a String
C++ Program to Find the Number of Vowels, Consonants, Digits and White Spaces in a String
C++ Program to Remove all Characters in a String Except Alphabet
C++ Program to Find the Length of a String
C++ Program to Concatenate Two Strings

C++ Strings



There are two types of strings commonly use in C++ programming language:

- Strings that are objects of string class (The Standard C++ Library string class)
- C-strings (C-style Strings)

C-strings

In C programming, only one type of string is available and this is also supported in C++ programming. Hence it's called C-strings. C-strings are the arrays of type `char` terminated with null character, that is, `\0` (ASCII value of null character is 0). Consider this example:

```
char str[] = "C++";
```

In the above code, `str` is a string and it holds 4 character. Although, "C++" has 3 character, the null character `\0` is added to the end of the string automatically.

 Strings Buffer in C++ Programming

Other ways of defining the string:

```
char str[4] = "C++";

char str[] = {'C','+','+','\0'};

char str[4] = {'C','+','+','\0'};
```

Like arrays, it is not necessary to use all the space allocated for the string. For example:

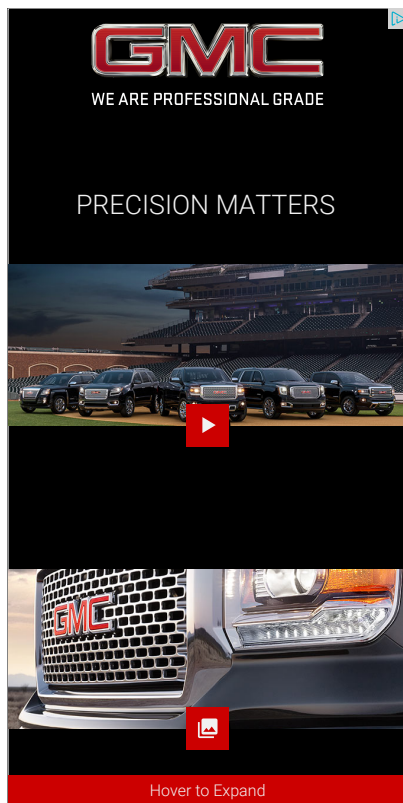
```
char str[100] = "C++";
```

Example 1: C++ String

C++ program to display a string entered by user.

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

int main() {
    char str[100];
    cout<<"Enter a string: ";
    cin>>str;
    cout<<"You entered: "<<str<<endl;
    cout<<"\nEnter another string: ";
    cin>>str;
```



```
cout<<"You entered: "<<str<<endl;
return 0;
}
```

Output

```
Enter a string: C++
You entered: C++

Enter another string: Programming is fun.
You entered: Programming
```

Notice that, in second example only "programming" is displayed instead of "Programming is fun". It is because The extraction operator >> considers a space has a terminating character.

Example 2: C++ String

C++ program to read and display an entire line entered by user.

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

int main() {
    char str[100];
    cout<<"Enter a string: ";
    cin.get(str, 100);
    cout<<"You entered: "<<str<<endl;
    return 0;
}
```

Output

```
Enter a string: Programming is fun.
You entered: Programming is fun.
```

To read the text containing blank space, `cin.get` function can be used. This function takes two arguments. First argument is the name of the string(address of first element of string) and second argument is the maximum size of the array.

Passing String to a Function

Strings are passed to a function in a similar way [arrays are passed to a function](#).

```
#include <iostream>
using namespace std;
void display(char s[]);

int main() {
    char str[100];
    cout<<"Enter a string: ";
    cin.get(str, 100);
    display(str);
    return 0;
}

void display(char s[]) {
    cout<<"You entered: "<<s;
}
```

Output

```
Enter a string: Programming is fun.
You entered: Programming is fun.
```



[About Us](#) | [Contact Us](#) | [Advertise With Us](#)

Copyright © by Programiz | All rights reserved | [Privacy Policy](#)