

# C++ Strings

In C++, string is an object of **std::string** class that represents sequence of characters. We can perform many operations on strings such as concatenation, comparison, conversion etc.

## C++ String Example

Let's see the simple example of C++ string.

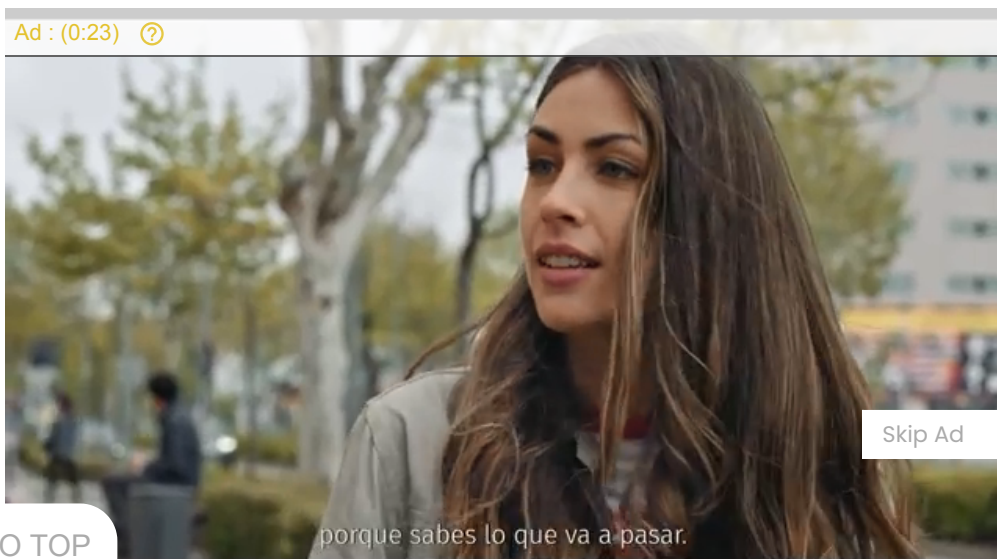
```
#include <iostream>
using namespace std;
int main() {
    string s1 = "Hello";
    char ch[] = { 'C', '+', '+' };
    string s2 = string(ch);
    cout<<s1<<endl;
    cout<<s2<<endl;
}
```

Output:

```
Hello
C++
```

## C++ String Compare Example

Let's see the simple example of string comparison using strcmp() function.



↑ SCROLL TO TOP

```
#include <iostream>
#include <cstring>
using namespace std;
int main ()
{
    char key[] = "mango";
    char buffer[50];
    do {
        cout<<"What is my favourite fruit? ";
        cin>>buffer;
    } while (strcmp (key,buffer) != 0);
    cout<<"Answer is correct!!"<<endl;
    return 0;
}
```

Output:

```
What is my favourite fruit? apple
What is my favourite fruit? banana
What is my favourite fruit? mango
Answer is correct!!
```



**Game Design Award  
2022**

iifamultimedia.in

## C++ String Concat Example

Let's see the simple example of string concatenation using strcat() function.

```
...>
```

↑ SCROLL TO TOP

```
using namespace std;

int main()
{
    char key[25], buffer[25];
    cout << "Enter the key string: ";
    cin.getline(key, 25);
    cout << "Enter the buffer string: ";
    cin.getline(buffer, 25);
    strcat(key, buffer);
    cout << "Key = " << key << endl;
    cout << "Buffer = " << buffer<<endl;
    return 0;
}
```

Output:

```
Enter the key string: Welcome to
Enter the buffer string: C++ Programming.
Key = Welcome to C++ Programming.
Buffer = C++ Programming.
```

## C++ String Copy Example

Let's see the simple example of copy the string using strcpy() function.

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

int main()
{
    char key[25], buffer[25];
    cout << "Enter the key string: ";
    cin.getline(key, 25);
    strcpy(buffer, key);
    cout << "Key = " << key << endl;
    cout << "Buffer = " << buffer<<endl;
    return 0;
}
```

↑ SCROLL TO TOP

Output:

```
Enter the key string: C++ Tutorial
Key = C++ Tutorial
Buffer = C++ Tutorial
```

## C++ String Length Example

Let's see the simple example of finding the string length using strlen() function.

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    char ary[] = "Welcome to C++ Programming";
    cout << "Length of String = " << strlen(ary)<<endl;
    return 0;
}
```

Output:

```
Length of String = 26
```

## C++ String Functions

Function	Description
int compare(const string& str)	It is used to compare two string objects.
int length()	It is used to find the length of the string.
void swap(string& str)	It is used to swap the values of two string objects.
string substr(int pos,int n)	It creates a new string object of n characters.
int size()	It returns the length of the string in terms of bytes.

↑ SCROLL TO TOP

<code>void resize(int n)</code>	It is used to resize the length of the string up to n characters.
<code>string&amp; replace(int pos,int len,string&amp; str)</code>	It replaces portion of the string that begins at character position pos and spans len characters.
<code>string&amp; append(const string&amp; str)</code>	It adds new characters at the end of another string object.
<code>char&amp; at(int pos)</code>	It is used to access an individual character at specified position pos.
<code>int find(string&amp; str,int pos,int n)</code>	It is used to find the string specified in the parameter.
<code>int find_first_of(string&amp; str,int pos,int n)</code>	It is used to find the first occurrence of the specified sequence.
<code>int find_first_not_of(string&amp; str,int pos,int n)</code>	It is used to search the string for the first character that does not match with any of the characters specified in the string.
<code>int find_last_of(string&amp; str,int pos,int n)</code>	It is used to search the string for the last character of specified sequence.
<code>int find_last_not_of(string&amp; str,int pos)</code>	It searches for the last character that does not match with the specified sequence.
<code>string&amp; insert()</code>	It inserts a new character before the character indicated by the position pos.
<code>int max_size()</code>	It finds the maximum length of the string.
<code>void push_back(char ch)</code>	It adds a new character ch at the end of the string.
<code>void pop_back()</code>	It removes a last character of the string.
<code>string&amp; assign()</code>	It assigns new value to the string.
<code>int copy(string&amp; str)</code>	It copies the contents of string into another.
<code>char&amp; back()</code>	It returns the reference of last character.
<code>Iterator begin()</code>	It returns the reference of first character.
<code>int capacity()</code>	It returns the allocated space for the string.
↑ SCROLL TO TOP 1()	It points to the first element of the string.

<code>const_iterator cend()</code>	It points to the last element of the string.
<code>void clear()</code>	It removes all the elements from the string.
<code>const_reverse_iterator crbegin()</code>	It points to the last character of the string.
<code>const_char* data()</code>	It copies the characters of string into an array.
<code>bool empty()</code>	It checks whether the string is empty or not.
<code>string&amp; erase()</code>	It removes the characters as specified.
<code>char&amp; front()</code>	It returns a reference of the first character.
<code>string&amp; operator+=()</code>	It appends a new character at the end of the string.
<code>string&amp; operator=()</code>	It assigns a new value to the string.
<code>char operator[] (pos)</code>	It retrieves a character at specified position pos.
<code>int rfind()</code>	It searches for the last occurrence of the string.
<code>iterator end()</code>	It references the last character of the string.
<code>reverse_iterator rend()</code>	It points to the first character of the string.
<code>void shrink_to_fit()</code>	It reduces the capacity and makes it equal to the size of the string.
<code>char* c_str()</code>	It returns pointer to an array that contains null terminated sequence of characters.
<code>const_reverse_iterator crend()</code>	It references the first character of the string.
<code>reverse_iterator rbegin()</code>	It reference the last character of the string.
<code>void reserve(inr len)</code>	It requests a change in capacity.
<code>allocator_type get_allocator();</code>	It returns the allocated object associated with the string.

[< Prev](#)
[Next >](#)
[↑ SCROLL TO TOP](#)