



≡ Menu



≡ View All Topics



Ads by Google

Stop seeing this ad

Why this ad? ⓘ

String Functions in C++: There is a different type of functions on strings. They are as follows:

- Input Functions
- Capacity Functions
- Iterator Functions
- manipulating Functions

And these functions of strings are again divided into sub-categories. Let us learn in detail regarding each function.

1) Input String Functions in C++

The input functions of strings are again divided into three different types. They are as follows:

- **getline():** It is used to store a stream of characters as entered by the user in the object memory.
- **push_back():** It is used to input a character at the end of the string.
- **pop_back():** It is used for deleting the last character from the string.



```
4.  int main()
5.  {
6.      //Declaring string
7.      string str;
8.      //Taking string input using getline() "freshersnow" in givin output
9.      getline(cin,str);
10.     //Displaying string
11.     cout<<"The initial string is: ";
12.     cout<<str<<endl;
13.     //Using push_back() to insert a character at the end pushes 's' in this case
14.     str.push_back('s');
15.     //Displaying string
16.     cout<<"The string after push_back operation is : ";
17.     cout<<str<<endl;
18.     //Using pop_back() to delete a character from end pops 's' in this case
19.     str.pop_back();
20.     //Displaying string
21.     cout<<"The string after pop_back operation is: ";
22.     cout<<str<<endl;
23.     return 0;
24. }
```

Output:

The initial string is: freshersnow

The string after push_back operation is: freshersnows

✓The string after non back operation is: freshersnow

- **capacity()**: It returns the capacity allocated to the string, which can be equal to (or) more than the size of the string. And also additional space is allocated so that when new characters are added to the string, the operation can be done directly.
- **resize()**: changes the size of string. i.e. increase (or) decrease
- **shrink_to_fit()**: It decreases the capacity of the string and makes it's equal to its size.

Example:

```
1. #include<iostream>
2. #include<string> //for string class
3. using namespace std;
4. int main()
5. {
6.     //Initializing string
7.     string str = "freshersnow is for freshers";
8.     //Displaying string
9.     cout<<"The initial string is : ";
10.    cout<<str << endl;
11.    //Resizing string using resize()
12.    str.resize(13);
13.    //Displaying string
14.    cout<<"The string after resize operation is : ";
15.    cout<< str << endl;
16.    //Displaying capacity of string
17.    cout<<"The capacity of string is : ";
18.    cout<<str.capacity() << endl;
```

```
24.     cout<<str.capacity() << endl;  
25.     return 0;  
26. }
```

Output:

The initial string is: freshersnow is for freshers

The string after resize operation is: freshersnow

The capacity of string is: 27

The new capacity after shrinking is: 15

3) Iterator String Functions

The iterator functions are again divided into subcategories. They are as follows:

- **begin()**: returns the iterator to the beginning of the string.
- **end()**: returns an iterator to end of the string
- **rbegin()**: It returns the reverse iterator pointing to end of the string.
- **rend()**: It returns the reverse iterator pointing at the beginning of a string.

Example:

```
1. #include<iostream>  
2. #include<string>//for string class  
3. using namespace std;
```

```
9.     std::string::iterator it;
10.    //Declaring reverse iterator
11.     std::string::reverse_iterator it1;
12.    //Displaying string
13.     cout<<"The string using forward iterators is : ";
14.     for(it=str.begin();it!=str.end(); it++)
15.         cout<<*it;
16.         cout<<endl;
17.    //Displaying reverse string
18.     cout<<"The reverse string using reverse iterators is : ";
19.     for(it1=str.rbegin();it1!=str.rend(); it1++)
20.         cout<<*it1;
21.         cout<<endl;
22.     return 0;
23. }
```

Output:

The string using forward iterators is: freshersnow

The reverse string using reverse iterators is: wonserhserf

4) Manipulating String Functions

The manipulating functions are also again divided into subcategories. They are as follows:

- ✓ • **copy("char array", len, pos):** The function copies the substring in the character array mentioned in its arguments.

```
1. #include<iostream>
2. #include<string> //for string class
3. using namespace std;
4. int main()
5. {
6.     //Initializing 1st string
7.     string str1 = "freshersnow is for freshers";
8.     //Declaring 2nd string
9.     string str2 = "freshersnow tutorial";
10.    //Declaring character array
11.    char ch[80];
12.    //using copy() to copy elements into char array copies "freshersnow"
13.    str1.copy(ch,13,0);
14.    //Displaying char array
15.    cout<<"The new copied character array is : ";
16.    cout<<ch<<endl<<endl;
17.    //Displaying strings before swapping
18.    cout<<"The 1st string before swapping is : ";
19.    cout<<str1<<endl;
20.    cout<<"The 2nd string before swapping is : ";
21.    cout<<str2<<endl;
22.    //using swap() to swap string content
23.    str1.swap(str2);
24.    //Displaying strings after swapping
25.    cout<<"The 1st string after swapping is : ";
26.    cout<<str1 << endl;
27.    cout<<"The 2nd string after swapping is : ";
```

The new copied character array is: freshersnow

The 1st string before swapping is: freshersnow is for freshers

The 2nd string before swapping is: freshersnow tutorial

The 1st string after swapping is: freshersnow tutorial

The 2nd string after swapping is: freshersnow is for freshers

List of String Functions in C++

Function	Description
int compare(const string& str)	compare two string objects
int length()	find the length of the string
void swap(string& str)	swap the values of two string objects
string substr(int pos,int n)	creates a new string object of n characters
int size()	Returns the length of the string in terms of bytes
void resize(int n)	Used to resize the length of the string up to n characters
string& replace(int pos,int len,string& str)	Replaces portion of the string that begins at character position pos and spans len characters
string& append(const string& str)	Adds new characters at the end of another string object

<code>int find_first_of(string& str,int pos,int n)</code>	used to find the first occurrence of the specified sequence
<code>int find_first_not_of(string& str,int pos,int n)</code>	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& str,int pos,int n)</code>	used to search the string for the last character of specified sequence
<code>int find_last_not_of(string& str,int pos)</code>	searches for the last character that does not match with the specified sequence
<code>string& insert()</code>	inserts a new character before the character indicated by the position pos
<code>int max_size()</code>	finds the maximum length of the string.
<code>void push_back(char ch)</code>	adds a new character ch at the end of the string
<code>void pop_back()</code>	removes a last character of the string
<code>string& assign()</code>	assigns new value to the string
<code>int copy(string& str)</code>	copies the contents of string into another
<code>char& back()</code>	returns the reference of last character
<code>Iterator begin()</code>	returns the reference of first character



<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>	checks whether the string is empty or not
<code>string& erase()</code>	It removes the characters as specified
<code>char& front()</code>	It returns a reference of the first character.
<code>string& operator+=()</code>	appends a new character at the end of the string
<code>string& operator=()</code>	assigns a new value to the string
<code>char operator[](pos)</code>	retrieves a character at specified position pos
<code>int rfind()</code>	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



```
allocator_type get_allocator();
```

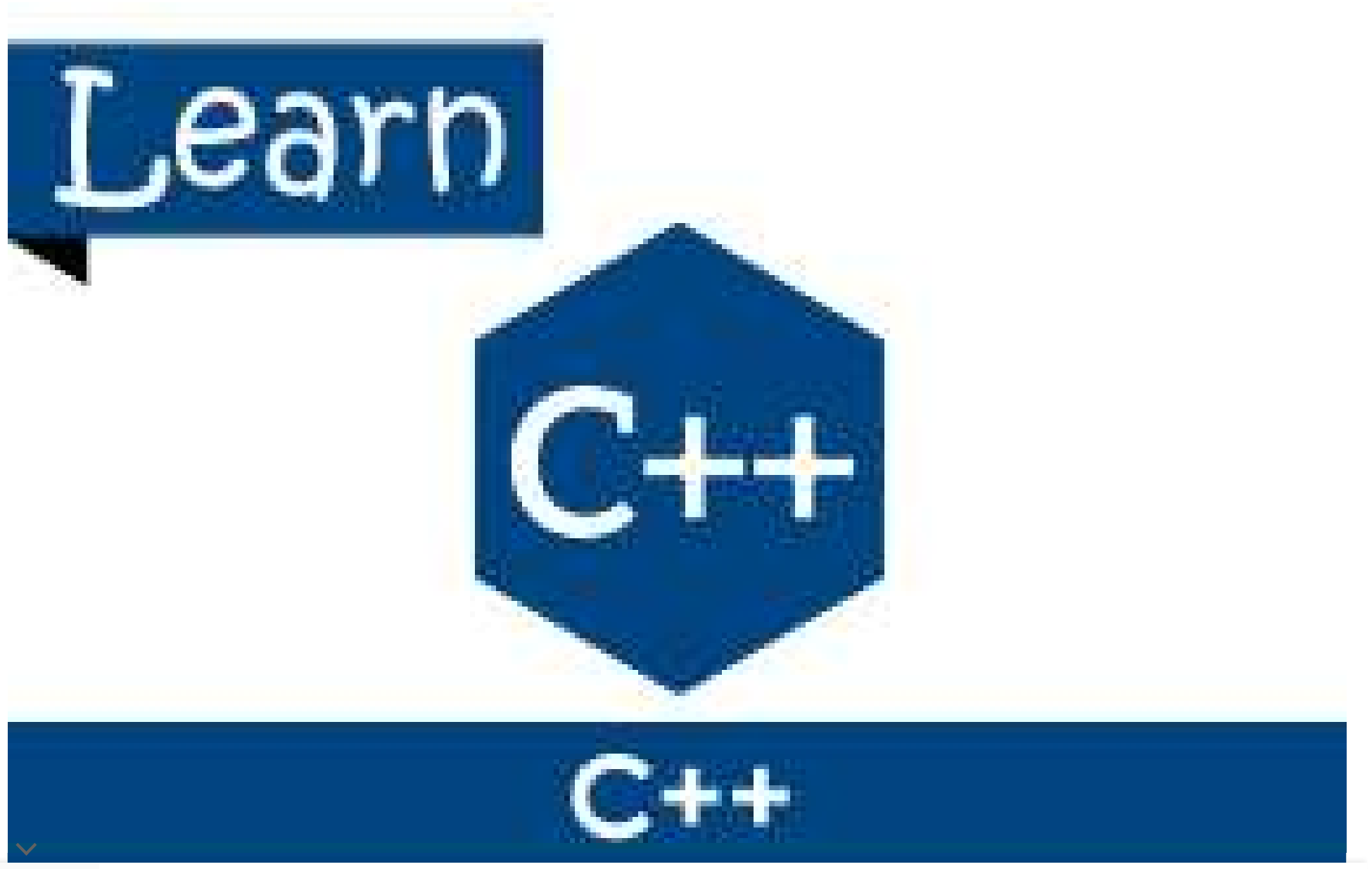
It returns the allocated object associated with the string

[← Previous Page](#)[Next Page →](#)

ALLOCATION AND DEALLOCATION







C++ Tutorial Tutorial

C++ Introduction

Difference Between C and C++

C++ Installation and setup

C++ Features

Streams in C++

Comments in C++

Structure of a C++ Program

C++ Manipulators

C++ Tokens

▼ Basic C++ Input Output

C++ Variables

C++ Keywords

Operators in C++

C++ Constants/ Literals

Variable Scope in C++

Storage Classes in C++

C++ Decision Control Statements

C++ if statement

C++ if else statement

C++ Nested if Statements

✓ C++ Switch Statements

C++ While loop

C++ Do-While Loop

C++ Infinite Loop

C++ Nested Loops

C++ Functions

Member Functions in C++

Parameter Passing References in C++ Functions

Difference Between Call by Value and Call by Reference in C++

C++ Function Overloading

C++ Function Overriding

✓ Function Overloading vs Function Overriding in C++

C++ Strings

String Functions in C++

String Operations in C++

String manipulation in C++

Pointers in C++

Array of Pointers in C++

Pointer to Pointer in C++

Difference between Arrays And Pointers in C++

C++ Pointer Arithmetic

C++ this Pointer

✓ OOPs Concept in C++

C++ Overloading

C++ Inheritance

Types of Inheritance in C++

Friend Class and Friend Function in C++

C++ Polymorphism

C++ Abstraction

C++ Encapsulation

C++ Structs

C++ Enumeration (enum)

Exception Handling in C++

✓ try, catch, throw Exceptions in C++

C++ Files and Streams

Namespaces in C++

C++ Templates

C++ Preprocessor Directives



B.TECH @ PES UNIVERSITY
ADMISSIONS CLOSING SOON



PES
All India Online Entrance
for Admission to PES Enclave

REGISTER TODAY



Tutorials

[AngularJS](#)[AWS Tutorial](#)[Big Data Tutorial](#)[C Language Tutorial](#)[C++ Tutorial](#)[CSS Tutorial](#)[Ethical Hacking Tutorial](#)[Hadoop Tutorial](#)[HTML Attributes](#)[HTML Tags](#)[HTML Tutorial](#)[HTML5 Tutorial](#)[Map Reduce Tutorial](#)[NodeJS Tutorial](#)[Python Tutorial](#)



- ★ RRB NTPC Stage 1 Mock Test
- ★ Banking Mock Tests
- ★ IT Software Mock Tests
- ★ SSC Mock Tests
- ★ Insurance Mock Tests
- ★ Accenture Mock Tests
- ★ Infosys Mock Tests
- ★ Tech Mahindra Mock Tests
- ★ Cognizant Mock Tests
- ★ HCL Mock Tests
- ★ Deloitte Mock Tests
- ★ TCS NQT Mock Tests

Preparation Materials

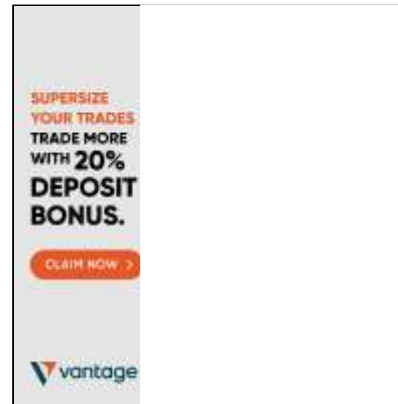
[Aptitude Questions](#)

[Reasoning Questions](#)

[GK Questions](#)

[English Questions](#)

[Placement Papers](#)



Jobs in India
▼

[Internship Jobs](#)[Startup Jobs](#)

Company

- ★ [About Us](#)
- ★ [Contact Us](#)
- ★ [Privacy Policy](#)
- ★ [Terms & Conditions](#)
- ★ [We Are Hiring](#)

Address:

Freshersnow, D.No 40-7/3-7/1

Tikkle Road, Labbipet, Vijayawada, Andhra Pradesh. ★ [Contact Us](#)

Popular Tutorials

★ [HTML Tutorial](#)

✓ ★ [CSS Tutorial](#)

★ [C++ Tutorial](#)

Latest Tutorials

★ [AWS Tutorial](#)

★ [AngularJS Tutorial](#)

★ [NodeJS Tutorial](#)

★ [ReactJS Tutorial](#)

★ [HTML5 Tutorial](#)

© 2023 FreshersNow Tutorials - Learn Free Courses Online • Powered by GeneratePress

