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.

```
int main()
{
 //Declaring string
 string str;
//Taking string input using getline() "freshersnow" in givin output
 getline(cin,str);
//Displaying string
 cout<<"The initial string is: ";
 cout<<str<<endl;
//Using push_back() to insert a character at the end pushes 's' in this case
  str.push_back('s');
//Displaying string
    cout<<"The string after push_back operation is : ";
    cout<<str<<endl;
//Using pop_back() to delete a character from end pops 's' in this case
    str.pop_back();
//Displaying string
  cout<<"The string after pop_back operation is: ";
  cout<<str<<endl;
  return 0;
}
```

Output:

The initial string is: freshersnow
The string after push_back operation is: freshersnows
The string after pop_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:

```
#include<iostream>
#include<string> //for string class
using namespace std;
int main()
{
  //Initializing string
   string str = "freshersnow is for freshers";
  //Displaying string
  cout<<"The initial string is : ";
  cout<<str << endl;
  //Resizing string using resize()
  str.resize(13);
  //Displaying string
  cout<<"The string after resize operation is : ";
  cout<< str << endl;
  //Displaying capacity of string
  cout<<"The capacity of string is : ";
  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;
```

```
    std::string::iterator it;
   //Declaring reverse iterator
    std::string::reverse_iterator it1;
   //Displaying string
     cout<<"The string using forward iterators is : ";
     for(it=str.begin();it!=str.end(); it++)
     cout<<*it;
     cout<<endl;
//Displaying reverse string
     cout<<"The reverse string using reverse iterators is : ";
     for(it1=str.rbegin();it1!=str.rend(); it1++)
     cout<<*it1;
     cout<<endl;
     return 0;
}
```

Output:

The string using forward iterators is: freshersnow
The reverse string using reverse iterators is: wonsrehserf

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.

```cpp
#include<iostream>
#include<string> //for string class
using namespace std;
int main()
{
   //Initializing 1st string
   string str1 = "freshersnow is for freshers";
   //Declaring 2nd string
  string str2 = "freshersnow tutorial";
  //Declaring character array
   char ch[80];
  //using copy() to copy elements into char array copies "freshersnow"
    str1.copy(ch,13,0);
  //Displaying char array
   cout<<"The new copied character array is : ";
   cout<<ch<<endl<<endl;
  //Displaying strings before swapping
   cout<<"The 1st string before swapping is : ";
   cout<<str1<<endl;
   cout<<"The 2nd string before swapping is : ";
   cout<<str2<<endl;
  //using swap() to swap string content
   str1.swap(str2);
   //Displaying strings after swapping
   cout<<"The 1st string after swapping is : ";
   cout<<str1 << endl;
   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

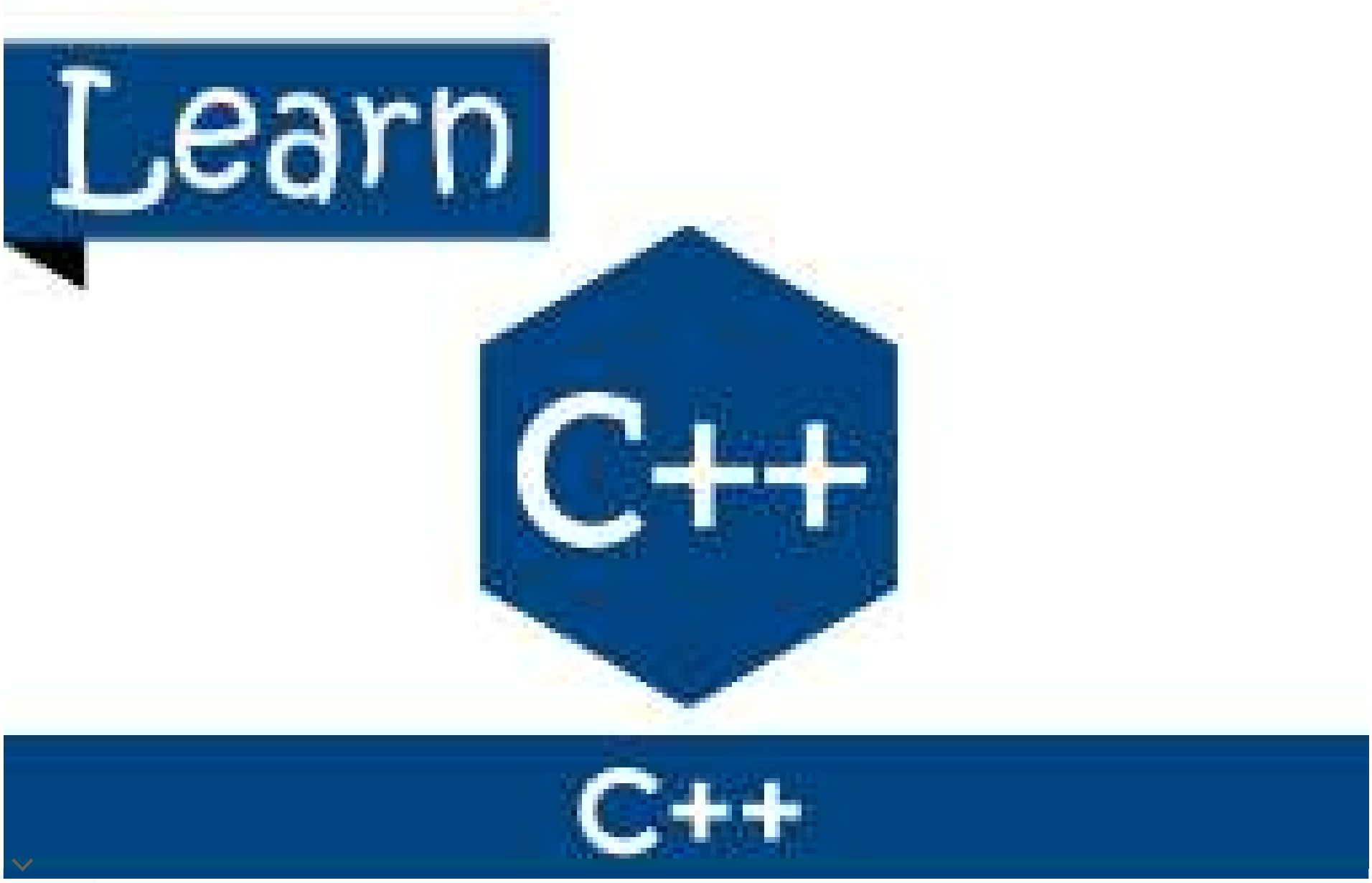
int find_first_of(string& str,int pos,int n)	used to find the first occurrence of the specified sequence
int find_first_not_of(string& str,int pos,int n)	used to search the string for the first character that does not match with any of the characters specified in the string
int find_last_of(string& str,int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

void clear( )	It removes all the elements from the string
const_reverse_iterator crbegin()	It points to the last character of the string
const_char* data()	It copies the characters of string into an array
bool empty( )	checks whether the string is empty or not
string& erase( )	It removes the characters as specified
char& front()	It returns a reference of the first character.
string& operator+=( )	appends a new character at the end of the string
string& operator=( )	assigns a new value to the string
char operator[](pos)	retrieves a character at specified position pos
int rfind( )	searches for the last occurrence of the string
iterator end( )	It references the last character of the string
reverse_iterator rend( )	It points to the first character of the string.
void shrink_to_fit()	It reduces the capacity and makes it equal to the size of the string
char* c_str()	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


Learn
C++
C++

C++ Tutorial Tutorial

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++ 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

Search ...

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

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