
C++ String

C++ String Manipulation

In C++, there are many functions that can be used to manipulate string objects. In this topic we will discuss some C++ string manipulation functions such as `size()`, `assign()`, `append()`, `insert()`, `compare()`, `erase()`, `find()`, `replace()`, `substr()`, and `empty()`.

C++ `size()` function

C++ provides the `size(string)` function to count the number of characters in a string object. This function accepts a string argument that you want to count its characters and returns the number of characters count.

Example:

```
#include <cstdlib>
#include <iostream>
#include <conio.h>

using namespace std;
void strmanip(){
string str="C++ programming code";
cout<<"The number of characters in the string is:"<<size(str);

}

int main(){

strmanip();

getch();

return 0;

}
```

C++ `assign()` function

The `assign()` function is used to assign all characters of a string to another string or just part of a string to another string. Normally, you can assign one string to another by using assignment operator(=). However, if you want to assign a proportion of a string to another string you need to use the `assign()` function instead. The `assign()` function has two forms. The first form is to assign all characters from a string to another and the second form assigns part of a string to another string.

```
assign(string);
```

```
assign(string, start, num_chars);
```

The second form of the `assign()` function accepts three arguments. The first argument is a string that you want to cut part of it and the start argument indicates where you want to start cutting. The `num_chars` argument is about the number of characters that you want to cut.

Example:

```
#include <cstdlib>
#include <iostream>
#include <conio.h>

using namespace std;

void strmanip(){
string str1;

string str2="C++ programming code";
str1.assign(str2);//assign entire string str2 to another string str1
cout<<str1<<endl;
str1.assign(str2,4,11);//assign part of string str 2 to another string str1
cout<<str1<<endl;

}

int main(){

strmanip();

getch();

return 0;

}
```

C++ `append()` function

The `append()` function is used to add a string to the end of another string. The `append()` function has two forms. The first form appends all characters in a string to another string. The second form appends part of string to another string. This proportion of string is taken out by starting from the start argument and the `num_chars` arguments specifies the number of characters to be taken out.

append(string);

append(string, start, num_chars);

Example:

```
#include <cstdlib>
#include <iostream>
#include <conio.h>

using namespace std;

void strmanip(){

string str="C++ programming";
str.append(" in easy steps");//append entire string " in easy steps" to another string str
cout<<str<<endl;
str.append(" in easy steps",3,5);//append part of string " in easy steps" to another string str
cout<<str<<endl;

}

int main(){

strmanip();

getch();

return 0;

}
```

C++ insert() function

By using insert() function you can insert a string or a proportion of a string to another string in a position that you want. The insert() function has two forms. The first form inserts all characters of a string to another string in the position specified by the *position* argument. The second form inserts a proportion of a string to another string.

```
insert(position, string);
```

```
insert(position,string, start, num_chars);
```

Example:

```
#include <cstdlib>
#include <iostream>
#include <conio.h>
```

```
using namespace std;
```

```
void strmanip(){
string str="C++ programming";

str.insert(3," Java");//insert string " Java" to string str at the position 3
cout<<str<<endl;
str.insert(3," C# programming",0,3);//insert string " C#" to string str at the position 3
cout<<str<<endl;

}

int main(){

strmanip();

getch();

return 0;

}
```

C++ compare() function

The compare() function is used to two strings. The compare() function has two forms. The first form compares all characters in a string with another string. The second form compares part of string with another string. This proportion of string is taken out by starting from the start argument and the num_chars argument specifies the number of characters to be taken out. This function returns less than zero if the invoking string is less than the string specified as an argument. It returns zero if both strings are equal.

```
compare(string);
```

```
compare(start, num_chars, string);
```

Example:

```
#include <cstdlib>
#include <iostream>
#include <conio.h>
```

```
using namespace std;
```

```
void strmanip(){
```

```
string st1,st2;
cout<<"Enter your first string:";
cin>>st1;
cout<<"Enter your second string:";
cin>>st2;
if(st1.compare(st2)>0) cout<<"First string is greater than second string."<<endl;
else if(st1.compare(0, st2.size(),st2)<0) cout<<"First string is less than second string."<<endl;
else cout<<"Both strings are equal."<<endl;

}

int main(){

strmanip();

getch();

return 0;

}
```

C++ erase() function

The erase(start, [end]) function is used to remove characters from a string. If the end position is not specified it will remove characters in a string by starting from the start argument till the end of the string.

Example:

```
#include <cstdlib>
#include <iostream>
#include <conio.h>

using namespace std;

void strmanip(){

string str="C++ programming code";
str.erase(4,11);// remove "programming" from string str
cout<<str<<endl;

}

int main(){

strmanip();
```

```
getch();  
  
return 0;  
  
}
```

C++ find() function

You can use the `find(string,[position])` function to find a substring in another string. The *position* argument specified the starting position of finding process in the invoking string. If you do not specify this argument the finding process will start at the beginning of the invoking string. The *string* argument is the string that you are looking for in the invoking string. This function returns the position of the matched string in the invoking string. If the target string is not found it returns -1.

Example:

```
#include <cstdlib>  
#include <iostream>  
#include <conio.h>  
  
using namespace std;  
  
void strmanip(){  
  
string str="C++ code programming";  
int i=str.find("code",3); //find string "code" in string str starting from position 3  
  
if(i!=-1)  
cout<<"\ncode was found at:"<<i<<endl;  
  
else cout<<"\ncode was not found"<<endl  
  
}  
  
int main(){  
  
strmanip();  
  
getch();  
  
return 0;  
  
}
```

C++ replace() function

You can use the `replace()` function to replace a substring by a new string in an invoking string. The `replace()` function has two forms. The first form can be used to replace a substring (starting from start argument and expanding for `num_chars` argument) in the invoking string by the entire content of the new string. The second form only takes a proportion of the new string content to replace the substring.

```
replace(start,num_chars,new_string);
```

```
replace(start, num_chars, new_string, start_in_new_string, num_chars);
```

Example:

```
#include <cstdlib>
#include <iostream>
#include <conio.h>
```

```
using namespace std;
```

```
void strmanip(){
```

```
string str="C++ code";
```

```
str.replace(4,4,"Programming");//replace "code" with "Programming"
```

```
cout<<str<<endl;
```

```
str.replace(0,3,"C Programming",0,1);//replace "C++" with "C"
```

```
cout<<str<<endl;
```

```
}
```

```
int main(){
```

```
strmanip();
```

```
getch();
```

```
return 0;
```

```
}
```

C++ substr() and empty functions

You can use `substr(start, num_chars)` function to get a proportion in `num_chars` length with the starting point specified as the start argument of a string. To check whether a string is empty before taking part of the string out, you can use `empty()` function.

Example:

```
#include <cstdlib>
#include <iostream>
#include <conio.h>

using namespace std;
void strmanip(){
string name;
cout<<"Enter your full name separating by underscore:";
cin>>name;
if(!name.empty()){

//find the position of "_"
int pos_space=name.find("_");

//take first name out
cout<<"Your first name is:"<<name.substr(0,pos_space)<<endl;

//take last name out
cout<<"Your last name is:"<<name.substr(pos_space+1,name.size()-pos_space)<<endl;

}
else cout<<"Your name is empty."<<endl;

}

int main(){

strmanip();

getch();

return 0;

}
```