



1

Library in c++ (as cmath , iostream ,
etc.)

2

Array of characters

3

Can contain small, capital
letters , numbers and symbols

4

Terminated by null character ('\0')

5

#include <string>

6

Can be treated as integers

String

is a sequence of characters ending with NULL '\0'

```
char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
```

If you follow the rule of array initialization, then you can write the above statement as follows:

```
char greeting[] = "Hello";
```

Following is the memory presentation of above defined string in C++:

Actually, you do not place the null character at the end of a string constant. The C++ compiler automatically places the '\0' at the end of the string when it initializes the array.

String declaration:

before declaring a string we need to call the string header library as following:

```
#include<string>
```

```
char a[4]="abc"; // == char a[4]={'a','b','c','\0'};
```

```
String x ("highschool") ;
```

```
String x = "highschool" ;
```

```
String x ; x = "highschool" ;
```

Operations on strings

Operator	Working
=	Assignment
+	Joining two or more strings
+=	Concatenation and assignment
==	Equality
!=	Not equal to
< , >	Less than , greater than
<=	Less than or equal
>=	Greater than or equal
[]	subscription

String Input:

We can use the (**cin**) to input string as any other data types:

```
cin>>s;
```

```
cin>>s[i] ;
```

Example:

```
#include<iostream>
```

```
#include<string>
```

```
Main(){ char word[80] ;
```

```
Do{ cin >>word ;
```

```
Cout << "\t" << word << endl ; }
```

```
while (word!=0)}
```

Example:

a program to read a string and print its elements:

```
#include <iostream>
#include <string>
int main()
{
    char s[] = "ABCDEF" ;
    for (int i=0;i<7;i++)
        cout<<"s["<<i<<"]=" "<<s[i]<<endl;
}
```

Output : s[0]='A'
 s[1]='B'
 s[2]='C'
 and so on.

String functions

Most common :

- 1- length
- 2- Copy
- 3- Concatenation
- 4- Compare
- 5- Assignment

Assignment (Assign())

```
#include <iostream>
#include <string.h>
using namespace std ;
int main ()
{
    string s1 ("c plus plus ");
    string s2 ;
    s2.assign (s1 , 0 , 6) ;
    cout << s2 ;
    return 0;
}
```

Out put : c plus

Length function (strlen)

strlen(str) used to find the length of string, i.e., the number of characters excluding the null terminator.

Ex.: strlen("ABC")

output : 3 (integer number).

Ex.: int length;
char a[]="XYZW";
length=strlen(a);

output: 4.

```
#include <iostream>
```

```
#include <string>
```

```
Using namespace std ;
```

```
int main ()
```

```
{
```

```
    //length () : determinates the length i.e. no. of characters
```

```
    string s1 ;
```

```
    cout << "length : " << s1.length () << endl ;
```

```
    s1 = " hello " ;
```

```
    cout << "length : " << s1.length () << endl ;
```

```
    return 0 ;
```

```
}
```

Output :

Length : 0

Length : 5

Example: write a c++ program to read a string and count the string length without using strlen() function:

Solution:

```
#include<iostream.h>
#include<string.h>
void main(){
char s[256];
int i,k;
cout<<"insert a string"<<endl;
cin>>s;
k=0;
for(i=0;s[i]!='\0';i++)
k++;
cout<<"length of the string is:"<<k<<endl;
}
```

Strcmp (compare function)

Comparison of Strings

The strcmp(str_1, str_2) function compares two strings and returns the following result :

0 : str_1 == str_2

positive number : str_1 > str_2

negative number : str_1 < str_2

*The strings are compared **lexicographically** (i.e., according to dictionary order):*

a < aa < aaa < ... < b < ba < bb < ... < bz < baa < ... < abca < abd < ...

comparing strings

```
#include <iostream>
#include <string.h>
using namespace std ;
int main ()
{
    char str [ 80 ] ;
    cout << " Enter your password : " ;
    gets (str) ;

    if (strcmp (str , "password" ) )           //strings differ
        cout << " invalid password " << endl ;
    else
        cout << " logged on " << endl ;

    return (0) ;

}
```

```
#include <iostream>
#include <string.h>
using namespace std ;
```

```
int main ()
{
    string s1 ("abd") ;
    string s2 ( "abc") ;
    int d = s1 . compare (s2) ;
    if ( d == 0 )
        cout << " \n Both strings are identical " ;
    else if ( d > 0 )
        cout << "\n s1 is greater than s2 " ;
    else
        cout << "\n s2 is greater than s1 " ;
    return 0 ;
}
```

Swap function (strswp)

Swap string exchange the given string with another string .

```
#include <iostream>
#include <string.h>
using namespace std ;
int main ()
{
    string buyer ("money");
    string seller ("goods");
    cout << "Before the swap, buyer has " << buyer;
    cout << " and seller has " << seller << '\n';
    swap (buyer,seller);
    cout << " After the swap, buyer has " << buyer;
    cout << " and seller has " << seller << '\n';
    return 0;
}
```

Out put :

Before the swap, buyer has money and seller has goods

After the swap, buyer has goods and seller has money

```
#include <string>
```

```
#include <iostream>
```

```
int main( )
```

```
{
```

```
using namespace std ;
```

```
// Declaring an object of type basic_string<char>
```

```
string s1 ( "Tweedledee" );
```

```
string s2 ( "Tweedledum" );
```

```
cout << "Before swapping string s1 and s2 : " << endl;
```

```
cout << "The basic_string s1 = " << s1 << "." << endl;
```

```
cout << "The basic_string s2 = " << s2 << "." << endl;
```

```
swap ( s1 , s2 );
```

```
cout << "\n After swapping string s1 and s2:" << endl;
```

```
cout << "The basic_string s1 = " << s1 << "." << endl;
```

```
cout << "The basic_string s2 = " << s2 << "." << endl;
```

```
}
```

Copy function (strcpy)

strcpy(s1,s2) : copy s2 instead of s1.

Example:

```
char s1[]="abc";  
char s2[]="xyz";  
strcpy(s1,s2);  
cout<<s1<<endl;  
cout<<s2<<endl;
```

output:

```
xyz  
xyz
```

strncpy(s1,s2,n): copy the first n characters of s2 into s1.

Example:

```
char s1[]="abcde";  
Char s2=[]="xyz";  
Strncpy(s1,s2,2);  
cout<<s1<<endl;  
cout<<s2<<endl;
```

output:

```
xycde  
xyz
```

Cocatenation function (strcat(),strncat())

strcat(s1,s2): insert s2 in the end of s1.
Strlen(s1) will be strlen(s1)+strlen(s2).

Example:

```
char s1[]="abc";  
Char s2=[]="xyz";  
Strcat(s1,s2);  
cout<<s1<<endl;  
cout<<s2<<endl;
```

output:

```
abcxyz  
xyz
```

strncat(s1,s2,n): insert n characters from s2 in the end of s1

Example:

```
char s1[]="abc";  
Char s2=[]="xyz";  
Strncat(s1,s2,2);  
cout<<s1<<endl;  
cout<<s2<<endl;
```

output:

```
abcxy  
xyz
```

```
#include <iostream>
#include <string>
using namespace std;
int main () {
string str1 = "Hello"; string str2 = "World"; string str3; int len ;
    str3 = str1;          // copy str1 into str3
cout << "str3 : " << str3 << endl;
str3 = str1 + str2;  // concatenates str1 and str2
cout << "str1 + str2 : " << str3 << endl;
len = str3.size(); // total length of str3 after concatenation
cout << "str3.size() : " << len << endl;
    return 0; }
```

Out put :

str3 : Hello

str1 + str2 : HelloWorld

str3.size() : 10

```
#include <iostream>
#include <string>
using namespace std;
int main () {
char str1[10] = "Hello"; char str2[10] = "World"; char str3[10]; int len ;
    strcpy( str3, str1);    // copy str1 into str3
cout << "strcpy( str3, str1) : " << str3 << endl;
    strcat( str1, str2);    // concatenates str1 and str2
cout << "strcat( str1, str2): " << str1 << endl;
    len = strlen(str1);    // total length of str1 after concatenation
cout << "strlen(str1) : " << len << endl;
    return 0; }
```

Out put:

strcpy(str3, str1) : Hello

strcat(str1, str2): HelloWorld strlen(str1) : 10

Example : write a C++ program to enter a matrix of 2-D string names and search for a name entered by keyboard to end the program when find it:

Solution:

```
#include<iostream>
#include<string.h>
void main(){
int i=0, t=0;
char myname [10];
char name[5][10]={"ali","huda","hadi","deyaa","samy"};
do{
cout<<"enter your name";
cin>>myname;
for (i=0;i<5;i++)
{
if(strcmp(myname, name[i])==0)
t=1;
}
while(t==0)
cout << " Welome " ;
}
```

Quizz

```
#include <iostream>
#include <string>
using namespace std;
int main ()
{
    string str ("Microsoft");
    for (size_t i = 0; i < str.length();)
    {
        cout << str.at(i-1);
    }
    return 0;
}
```

- a) M
- b) Microsoft
- c) Micro
- d) runtime error

Answer :

What is the out put ?

```
#include <iostream>
#include <cstring>
using namespace std;
int main ()
{
    char str1[10] = "Hello";
    char str2[10] = "World";
    char str3[10];
    int len ;
    strcpy( str3, str1);
    strcat( str1, str2)
    len = strlen(str1);
    cout << len << endl;
    return 0;
}
```

- a) 5
- b) 55
- c) 11
- d) 10

Out put :

What is the out put ?

```
#include <iostream>
#include <string>
using namespace std;
int main ()
{
    string str ("Ubuntu");
    cout << str.capacity();
    cout << str.max_size();
    return 0;
}
```

- a) 61073741820
- b) 51073741820
- c) 6 and max size depends on compiler
- d) none of the mentioned

Answer :