**C++ <string> example code:**

#include <string>  
#include <iostream>  
  
using namespace std;  
  
main()  
{  
 string a("abcd efg");  
 string b("xyz ijk");  
 string c;  
  
 cout << a << " " << b << endl; // Output: abcd efg xyz ijk  
  
 cout << "String empty: " << c.empty() << endl; // String empty: 1   
 // Is string empty? Yes it is empty. (TRUE)  
 c = a + b; // concatenation  
 cout << c << endl; // abcd efgxyz ijk  
 cout << "String length: " << c.length() << endl; // String length: 15  
 cout << "String size: " << c.size() << endl; // String size: 15  
 cout << "String capacity: " << c.capacity() << endl; // String capacity: 15  
 cout << "String empty: " << c.empty() << endl; // String empty: 0   
 // Is string empty? No it is NOT empty. (FALSE)  
 string d = c;  
 cout << d << endl; // abcd efgxyz ijk  
  
 // First character: a  
 cout << "First character: " << c[0] << endl; // Strings start with index 0 just like C.  
  
 string f(" Leading and trailing blanks ");  
 cout << "String f:" << f << endl;  
 cout << "String length: " << f.length() << endl; // String length: 37  
 cout << "String f:" << f.append("ZZZ") << endl; // String f: Leading and trailing blanks ZZZ  
 cout << "String length: " << f.length() << endl; // String length: 40  
  
 string g("abc abc abd abc");  
 cout << "String g: " << g << endl; // String g: abc abc abd abc  
 cout << "Replace 12,1,\"xyz\",3: " << g.replace(12,1,"xyz",3) << endl; // Replace 12,1,"xyz",3: abc abc abd xyzbc  
 cout << g.replace(0,3,"xyz",3) << endl; // xyz abc abd xyzbc  
 cout << g.replace(4,3,"xyz",3) << endl; // xyz xyz abd xyzbc  
 cout << g.replace(4,3,"ijk",1) << endl; // xyz i abd xyzbc  
 cout << "Find: " << g.find("abd",1) << endl; // Find: 6  
 cout << g.find("qrs",1) << endl;  
  
 string h("abc abc abd abc");  
 cout << "String h: " << h << endl;  
 cout << "Find \"abc\",0: " << h.find("abc",0) << endl; // Find "abc",0: 0  
 cout << "Find \"abc\",1: " << h.find("abc",1) << endl; // Find "abc",1: 4  
 cout << "Find\_first\_of \"abc\",0: " << h.find\_first\_of("abc",0) << endl; // Find\_first\_of "abc",0: 0  
 cout << "Find\_last\_of \"abc\",0: " << h.find\_last\_of("abc",0) << endl; // Find\_last\_of "abc",0: 0  
 cout << "Find\_first\_not\_of \"abc\",0: " << h.find\_first\_not\_of("abc",0) << endl; // Find\_first\_not\_of "abc",0: 3  
 cout << "Find\_first\_not\_of \" \": " << h.find\_first\_not\_of(" ") << endl; // Find\_first\_not\_of " ": 0  
 cout << "Substr 5,9: " << h.substr(5,9) << endl; // Substr 5,9: bc abd ab  
 cout << "Compare 0,3,\"abc\": " << h.compare(0,3,"abc") << endl; // Compare 0,3,"abc": 0  
 cout << "Compare 0,3,\"abd\": " << h.compare(0,3,"abd") << endl; // Compare 0,3,"abd": -1  
 cout << h.assign("xyz",0,3) << endl; // xyz  
 cout << "First character: " << h[0] << endl; // Strings start with 0 // First character: x  
}

**Output:**

abcd efg xyz ijk  
String empty: 1  
abcd efgxyz ijk  
String length: 15  
String size: 15  
String capacity: 15  
String empty: 0  
abcd efgxyz ijk  
First character: a  
String f: Leading and trailing blanks   
String length: 37  
String f: Leading and trailing blanks ZZZ  
String length: 40  
String g: abc abc abd abc  
Replace 12,1,"xyz",3: abc abc abd xyzbc  
xyz abc abd xyzbc  
xyz xyz abd xyzbc  
xyz i abd xyzbc  
Find: 6  
4294967295  
String h: abc abc abd abc  
Find "abc",0: 0  
Find "abc",1: 4  
Find\_first\_of "abc",0: 0  
Find\_last\_of "abc",0: 0  
Find\_first\_not\_of "abc",0: 3  
Find\_first\_not\_of " ": 0  
Substr 5,9: bc abd ab  
Compare 0,3,"abc": 0  
Compare 0,3,"abd": -1  
xyz  
First character: x