

REBOOT ACADEMY

Computer Training Institute

String Sample Code in C++

Index: Link to sample code

[string basics](#)

[length\(\)](#)

[resize\(\)](#)

[at\(\)](#)

[concatenation](#)

[find\(\)](#)

[substr\(\)](#)

[replace\(\)](#)

[compare\(\)](#)

[passing string to a function](#)

Note:

This is not a complete functional application these are sample codes, so run functions one by one or take care of the declaration and initialization of string variables to avoid unwanted errors.

```

#include<iostream>
#include<string> // class to deal with String
using namespace std;

void pass_string(string &name);
int main()
{
    //data type String
    string name;
    int a,n;
    cout<<"enter roll no.";
    cin>>a; //space and \n
    cout<<"Enter name";
    cin>>name;
    cout<<"Hello " <<name<<endl<<endl;
    //If name has spaces then use getline()
    /*
    cin.ignore(); // to flush out new line character and extra spaces
    cout<<"enter name";
    getline(cin, name); // gets terminated with \n
    cout<<"Hello " <<name<<endl<<endl;
    */

    //functions of string
    //1. length() function of the string
    cout<<"String Length Function"<<endl;
    cout<<"int length()"<<endl;
    n= name.length(); //parameter and return value
    cout<<"length= " <<n<<endl; //print length using variable

    cout<<"length= " <<name.length()<<endl; //print length directly
    //0 - length -1
    for(int i= 0; i<name.length();i++)
        cout<<name[i];
    cout<<endl<<endl;

    //2. resize() function of the string
    cout<<"String Resize Function"<<endl;
    cout<<"void resize(int n, char c)"<<endl;
    cout<<"void resize(int n)"<<endl;

    name.resize(n+3);
    cout<<name<<endl;
    name.resize(n+7, 'x');
  
```

```

cout<<name<<endl<<endl;

//3. at() function of the string
cout<<"String at() Function"<<endl;
cout<<"char at(int pos)"<<endl;

//for(int i= 0; i<name.length();i++)
    cout<<name.at(2)<<endl;

cout<<endl<<endl;
//4. Concatenation of the string
cout<<"String concatenation"<<endl;
cout<<"using operator +="<<endl;

cout<< endl<< "hello " + name;
string surname;
cout<<"enter surname";
cin>>surname;
name += surname; //concatenation of the string
cout<<endl<<name;
cout<<endl<<endl;

//5. find() function of the String
cout<<"String find() function"<<endl;
cout<<"int find ( char c, int pos = 0 ) const;"<<endl;
cout<<"int find ( string s, int pos = 0 ) const;"<<endl;
n = name.find("re",0); // returns index number if found else -1
cout<<n<<endl<<endl;

n= name.find_first_of("va",4); // it will check for "va" or "v" or "a"
starting from index 4
cout<<n<<endl<<endl;

//6. substr() function of the String
cout<<"String substr() function"<<endl;
cout<<"string substr( int pos, int n=whatIsLeft);"<<endl;
cout<<name.substr(1,6)<<endl; // returns string starting from index 1 to
5[6 will not be included]
    cout<<name.substr(3)<<endl;// returns string starting from index 3 till
the end of the string
    cout<<endl<<endl;

```

```
//7. replace() function of the String
cout<<"String replace() function"<<endl;
cout<<"string replace( int pos, int n, string str);"<<endl;
name.replace(2, 3 , "hello");// it will replace 3 characters starting from
index 2 with "hello"
cout<<name;
cout<<endl<<endl;

//8. compare() function of the String
cout<<"String compare() function"<<endl;
cout<<"string compare(string str);"<<endl
name = "Reboot";
string newstr= "reboot";
n =name.compare(newstr); //-1 if first string is smaller, 0 if they are
equal, 1 if second string is smaller
cout<<n;
cout<<endl<<endl;

//9. String should always be passed as reference as it is an object type

pass_string(name);
cout<<name<<endl;
cout<<endl<<"after Pass_string";
cout<<name<<endl;

return 0;
}

void pass_string( string &nn)
{
    cout<<"Passed String"<<nn;
    nn = nn + " kk";
}
```