

Sep 03, 2015

# COMPUTER PROGRAMMING

LAB 4

ABDUL AZIZ

Lab Instructor	Mr. Abdul Aziz
Course	Computer Programming Lab
Duration	2hrs

### Objectives:

In this lab, following topics will be covered

- ❖ Introduction to Strings
- ❖ Creating Strings
- ❖ Reading Strings
- ❖ String Constructors
- ❖ Most Common Function used in Strings
- ❖ Accessing characters in Strings
- ❖ String Streams

## 1. Introduction to Strings

String is actually a one-dimensional array of characters which is terminated by a **null** character '\0'. Thus a null-terminated string contains the characters that comprise the string followed by a **null**.

The following declaration and initialization create a string consisting of the word "Hello". To hold the null character at the end of the array, the size of the character array containing the string is one more than the number of characters in the word "Hello."

```
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++:

Index	0	1	2	3	4	5
Variable	H	e	l	l	o	\0
Address	0x23451	0x23452	0x23453	0x23454	0x23455	0x23456

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. Let us try to print above-mentioned string:

```
#include <iostream>
using namespace std;
int main ()
{
    char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};

    cout << "Greeting message: ";

    cout << greeting << endl;

    return 0;
}
```

When the above code is compiled and executed, it produces result something as follows:

Greeting Message: Hello

## 2. Creating Strings

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

int main()
{
    string first_name("Agha");
    string last_name = "khan";
    string ph_num(9,'1');

    cout<<"First Name: "<<first_name<<endl;
    cout<<"Last_Name: "<<last_name<<endl;
    cout<<"Phone Number: "<<ph_num;

    return 0;
}
```

### 3. Reading Strings

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

int main()
{
    string first_name;
    string last_name;
    cout<<"Enter First Name\n";
    cin>>first_name;
    cout<<"Enter Last Name\n";
    cin>>last_name;
    cout<<first_name<<" "<<last_name<<"\n";
    return 0;
}
```

### 4. String Constructors

```
#include<iostream>
#include<string>
using namespace std;
int main()
{
    string first_name = "Suleman";
    string last_name("Memon");
    string family_name(last_name);
    cout<<first_name<<" "<<last_name<<" "<<family_name;
    return 0;
}
```

C++ supports a wide range of functions that manipulate null-terminated strings:

S.N.	Function & Purpose
1	<b>strcpy(s1, s2);</b> Copies string s2 into string s1.
2	<b>strcat(s1, s2);</b> Concatenates string s2 onto the end of string s1.
3	<b>strlen(s1);</b> Returns the length of string s1.
4	<b>strcmp(s1, s2);</b> Returns 0 if s1 and s2 are the same; less than 0 if s1<s2; greater than 0 if s1>s2.
5	<b>Append()</b> Appends a part of the string to another string

**Here are the rest of functions with implementation.**

**i. getline()**

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

int main()
{
    string address;
    cout<<"Enter Address\n";
    getline(cin,address);
    cout<<"Address is "<<address;
    return 0;
}
```

## ii. Insert() and Erase()

```
int main()
{
    string str1 = "12345678";
    string str2 = "abcdefgh";
    cout<<str1<<endl;
    str1.insert(3,str2);           //123abcdefgh45678
    cout<<str1;
    cout<<"\n";
    str1.erase(3,8);             //12345678
    cout<<str1<<endl;
}
```

## iii. Swap()

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

int main()
{
    string str1 = "Bits", str2 = "Pieces";
    cout<<str1<<" and "<<str2<<"\n";
    str1.swap(str2);
    cout<<"After Swapping "<<endl<<str1<<" and "<<str2;
    return 0;
}
```

## iv. Length()

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

int main()
{
    string str = "DNA stands for Dioxyriboneuclicacid ";
    for(int i=0; i<str.length();i++)
    {
        cout<<str[i];
    }
    cout<<" "<<str.length();
}
```

## v. Substr()

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

int main()
{
    string str = "Dioxyriboneuclicacid";

    cout<<str.substr(5,11);
}
```

## 5. Accessing Characters in String

```
#include<iostream>
#include<string>
using namespace std;
int main()
{
    string str1 = "A quick brown fox";
    string str2 = "A Cat eats a mouse";

    for(int i =0; i<str2.length(); i++)
    {
        cout<<str1[i];
    }
    cout<<endl;

    int x = str1.find("quick");

    cout<<"quick is found at index "<<x<<endl;

    x = str2.find_last_of('a');
    cout<<"last occurence of a is at "<<x;
}
```

## 6. String Stream Processing (sstream)

So far, all of the I/O examples you have seen have been writing to cout or reading from cin. However, there is another set of classes called the stream classes for strings that allow you to use the familiar insertions (<<) and extraction (>>) operators to work with strings. Like istream and ostream, the string streams provide a buffer to hold data. However, unlike cin and cout, these streams are not connected to an I/O channel (such as a keyboard, monitor, etc...). **One of the primary uses of string streams is to buffer output for display at a later time, or to process input line-by-line.**

There are two ways to get data into a stringstream:

- 1) Use the insertion (<<) operator:

```
stringstream os;  
os << "en garde!" << endl; // insert "en garde!" into the stringstream
```

- 2) Use the str(string) function to set the value of the buffer:

```
stringstream os;  
os.str("en garde!"); // set the stringstream buffer to "en garde!"
```

There are similarly two ways to get data out of a stringstream: 1) Use the str() function to retrieve the results of the buffer:

```
1) stringstream os;  
   os << "12345 67.89" << endl;  
   cout << os.str();
```

This will print            **12345 67.89**

### 2) Use the extraction (>>) operator:

```
#include<iostream>  
#include<string>  
#include<sstream>  
using namespace std;  
int main()  
{  
    stringstream os;  
    os << "123456 789 67.89"; // insert a string of numbers into the stream  
  
    string strValue;  
    os >> strValue;  
  
    string strValue2;  
    os >> strValue2;  
  
    string strValue3;  
    os >> strValue3;
```



```

cout<<strValue<<endl<<strValue2<<endl<<strValue3;

}

```

OUTPUT will be:

```

123456
789
67.89

```

**Note that the >> operator iterates through the string -- each successive use of >> returns the next extractable value in the stream. On the other hand, str() returns the whole value of the stream, even if the >> has already been used on the stream.**

## 6.1 Clearing a stringstream for reuse

There are several ways to empty a stringstream's buffer.

- 1) Set it to the empty string using str():

```

stringstream os;
os << "Hello ";
os.str("");           // erase the buffer
os << "World!";
cout << os.str();

```

- 2) Call erase() on the result of str():

```

stringstream os;
os << "Hello ";

os.str(string());    // erase the buffer

os << "World!";
cout << os.str();

```

Both of these programs produce the following result:

**World!**

### Exercise

1. Write a program that asks user to write a sentence/paragraph and it displays the count of vowels and identify on which index they appeared in the given sentence/paragraph.

E.g.

**Input :** Long Live Pakistan

**Output:**

“a” = 2 times on index 10 & 15, “e” = 1 times on index 8, “i” = 2 times on index 6 & 12, “o” = 1 times on index 1, “u” = 0 times

2. Write a Login Program that reads actual password from input file and prompts user to enter a password either in small alphabet or in capital alphabet, then compares whether both (by user and from the file) are matching or not.

E.g

Password from input file : user1201

User Input : abcd

Output : Login Failed

3. Write a program by using string streams that displays the menu for shopping and asks the user to buy things accordingly. When user finishes the shopping, your program should display the receipt of bought items. You have to use sstream library.

E.g **Welcome to Imtiaz Super Market**

Item Name	Price
Clear Shampoo	160
Pepsi 1.5L	90
T-Shirt	750
Imtiaz Special Pen	115

OUTPUT:

You have bought the following items

ITEM NAME	PRICE	QUANTITY
Pepsi 1.5L	90	10
Imtiaz Special Pen	115	06
T-Shirt	750	04
Total		Rs. 4590/=