C++ Programming Language

Presented By:

T.A. Asmaa Hamad El-saied

E-mail: eng.asmaa134@gmail.com

Contents

Strings

- **C++ provides following two types of string representations:**
  - The C-style character string.
  - The string class type introduced with Standard C++.

Strings(cont.)

- **The C-Style Character String:**
  - The C-style character string originated within the C language and continues to be supported within C++. This 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.

Strings(cont.)

- **The C-Style Character String:**
  - The following declaration and initialization create a string consisting of 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";
```

  - 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.

Strings(cont.)

- **The C-Style Character 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

Strings(cont.)

- **The C-Style Character String:**
  - C++ supports a wide range of functions that manipulate null-terminated strings:

S.N.	Function & Purpose
1	strcpy(s1, s2); Copies string s2 into string s1.
2	strcat(s1, s2); Concatenates string s2 onto the end of string s1.
3	strlen(s1); Returns the length of string s1.
4	strcmp(s1, s2); Returns 0 if s1 and s2 are the same; less than 0 if s1<s2; greater than 0 if s1>s2.
5	strchr(s1, ch); Returns a pointer to the first occurrence of character ch in string s1.
6	strstr(s1, s2);

Strings(cont.)

- **The C-Style Character String:**

```
#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); // 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); // concatenates str1 and str2 cout << "strlen(str1) : " << len << endl; return 0; }
```

Output:

```
strcpy( str3, str1) : Hello
strcat( str1, str2): HelloWorld
strlen(str1) : 10
```

Strings(cont.)

- **The String Class in C++:**
  - The standard C++ library provides a string class type that supports all the operations mentioned above, additionally much more functionality.

Strings(cont.)

- **The String Class in C++:Example**

```
#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 lenghth of str3 after concatenation
       cout << "str3.size() : " << len << endl;
      return 0; }
```

Output:

```
str3 : Hello
str1 + str2 : HelloWorld
str3.size() : 10
```

Thanks