

Pointers

PART 3

Strings

A string is a series of characters treated as a single unit. A string may include letters, digits and various special characters such as +, -, *, /and \$. String literals, or string constants,

in C++ are written in double quotation marks as follows:

"John Q. Doe" (a name)

"9999 Main Street" (a street address)

"Maynard, Massachusetts" (a city and state)

"(201) 555-1212" (a telephone number)

Pointer-Based Strings

A pointer-based string is a built-in array of characters ending with a null character ('\0'), which marks where the string terminates in memory. A string is accessed via a pointer to its first character. The sizeof a string literal is the length of the string including the terminating null character. Pointer-based strings are like built-in arrays—a built-in array name is also a pointer to its first element.

String Literals as Initializers

A string literal may be used as an initializer in the declaration of either a built-in array of chars or a variable of type `const char *`. The declarations

```
char color[] = "blue";  
const char *colorPtr = "blue";
```

each initialize a variable to the string "blue". The first declaration creates a five-element built-in array `color` containing the characters 'b', 'l', 'u', 'e' and '\0'. The second declaration creates pointer variable `colorPtr` that points to the letter b in the string "blue" (which ends in '\0') somewhere in memory

Character Constants as Initializers

The declaration `char color[] = "blue";` could also be written

```
char color[] = { 'b', 'l', 'u', 'e', '\0' };
```

which uses character constants in single quotes (') as initializers for each element of the

built-in array. When declaring a built-in array of chars to contain a string, the built-in array must be large enough to store the string and its terminating null character. The compiler determines the size of the built-in array in the preceding declaration, based on the number of initializers in the initializer list

Accessing Characters in a C String

Because a C string is a built-in array of characters, we can access individual characters in a string directly with array subscript notation. For example, in the preceding declaration, `color[0]` is the character 'b', `color[2]` is 'u' and `color[4]` is the null character.

Reading Strings into char Built-In Arrays with cin

A string can be read into a built-in array of chars using stream extraction with cin. For example, the following statement reads a string into the built-in 20-element array of chars named word:

```
cin >> word;
```

The string entered by the user is stored in word. The preceding statement reads characters until a white-space character or end-of-file indicator is encountered. The string should be no longer than 19 characters to leave room for the terminating null character. The setw stream manipulator can be used to ensure that the string read into word does not exceed the size of the built-in array. For example, the statement

```
cin >> setw( 20 ) >> word;
```

The above statement specifies that cin should read a maximum of 19 characters into word and save the 20th location to store the terminating null character for the string.

If more than 19 characters are entered, the remaining characters are not saved in word, but they will be in the input stream and can be read by the next input operation.

Reading Lines of Text into char Built-In Arrays with cin.getline

In some cases, it's desirable to input an entire line of text into a built-in array of chars. For this purpose, the cin object provides the member function getline, which takes three arguments—a built-in array of chars in which the line of text will be stored, a length and a delimiter character. For example, the statements

```
char sentence[ 80 ];  
cin.getline( sentence, 80, '\n' );
```

Displaying C Strings

A built-in array of chars representing a null-terminated string can be output with cout and <<

```
cout << sentence;
```

Pointers and String

Strings can also be declared using pointers. Let's see an example.

```
#include <iostream>
using namespace std;
int main(){
    char name[ ]= "Sam";
    char *p;
    p = name;    /* for string, only this declaration will store its base address */
    while( *p != '\0')
    {
        cout << *p;
        p++;
    }
    return 0;
}
```

Built in functions

Function	Use
strlen	calculates the length of string
strcat	Appends one string at the end of another
strncat	Appends first n characters of a string at the end of another
strcpy	Copies a string into another
strncpy	Copies first n characters of one string into another
strcmp	Compares two strings
strncmp	Compares first n characters of two strings
strchr	Finds first occurrence of a given character in a string
strrchr	Finds last occurrence of a given character in a string
strstr	Finds first occurrence of a given string in another string