

Strings, and the `cout` Object in C++

Both the textbook and the Course Modules discuss character data, and mention both the ASCII (8-bit) and Unicode (16-bit) methods of representing character data in a computer. But only rarely do we deal with single characters. In most cases, we deal with several characters arranged into a string. Strings are discussed in the textbook on pages 17 and 18, and occasionally thereafter.

The purposes of this note are, first, to provide some additional items of information about strings, specifically the use of *escape characters*; and second, to explain the usage of the `cout` method found in the sample C++ program in the UMUC Programming Guide and in the Programming Exercises in the Course Modules. The first discussion will apply equally to the C++ and Java languages. It will do so by carefully avoiding the issue of how strings are stored in a computer, which is different for the two languages. The `cout` material is unique to C++.

A. Strings and Escape Characters

A string is a list of characters. When we need to refer to a specific string in a computer program, we enclose the characters of the string in quotation marks. Here are some examples of strings:

```
"PAUSE"  
"The sun was shining on the sea."  
"+-*&_567m"
```

That's simple enough, but it has the following problems:

- What if we want the character " to appear in a string?
- What if we want the string to contain several lines of text? We need some way to indicate where in the string the end of each line occurs, that is, a *newline*.
- What if we want some data arranged in neat columns? We need a way of representing a *tab*.

The way these problems are dealt with is that the backslash (`\`) is designated as an *escape character*. When the compiler encounters `\` inside a string, it examines the next character to see what special character is intended.

Here are the most common escape sequences:

Name of Character	The Character	How Written
backslash	<code>\</code>	<code>\\</code>
double quote	<code>"</code>	<code>\"</code>
single quote	<code>'</code>	<code>\'</code>
newline		<code>\n</code>
tab		<code>\t</code>

Here is an example. The represent the string

"The time has come," the Walrus said,
"To talk of many things."

Would be represented in a C++ by

```
"\t\"The time has come,\" the Walrus said,\n\t\"To talk of many things\""
```

in which the escape sequences have been colored to make them more conspicuous.

B. cout

`cout` is a feature of the C++ language that make it very easy to write statements for console output. Syntactically, it is quite different from all the rest of the language. So don't even think of looking for similarities to anything else in the language (or any other programming language). But, it will serve us very well in this course, because it is so easy to use.

The general form is:

```
cout << expression1 << expression2 << ... << expressionN;
```

This statement causes *expression1* through *expressionN* to be converted to a character strings and the resulting strings displayed on the console. Any of the *expressions* can be a string, or an arithmetic expression, or `endl`, the latter indicating a newline character. If any *expression* is an arithmetic expression, the arithmetic expression is evaluated and the resulting numeric value is converted to a character string for display.

Here are some examples

Program Text	Discussion
<pre>cout << i;</pre>	Convert the value of the integer variable <code>i</code> to a character string, and display the resulting character string on the console.
<pre>cout << "\t";</pre>	Send a tab character to the console.
<pre>cout << "The height is " << x + y << " feet." << endl;</pre>	Display "The height is " followed by the sum of <code>x</code> and <code>y</code> followed by "feet." and a newline. The numerical sum of <code>x</code> and <code>y</code> is converted to a character string for display.

There are more examples in the sample programs.