

Supplement V.D: printf Statements

For Introduction to C++ Programming
By Y. Daniel Liang

In the C language, you use the `printf` statement to display formatted output. You can also use it in the C++ language. But, you should use the `cout` object with stream manipulators to format output in C++, rather than the `printf` statements.

You may still see the `printf` statements in some C++ programs. The `printf` statement formats the output using following syntax:

```
printf(format, item1, item2, ..., itemk)
```

<Side Remark: specifier>

where `format` is a string that may consist of substrings and format specifiers. A format specifier specifies how an item should be displayed. An item may be a numeric value, a character, a boolean value, or a string. Each specifier begins with a percent sign. Table 1 lists some frequently used specifiers:

Table 1
Frequently Used Specifiers

Specifier	Output	Example
<code>%b</code>	a boolean value	true or false
<code>%c</code>	a character	'a'
<code>%d</code>	a decimal integer	200
<code>%f</code>	a floating-point number	45.460000
<code>%e</code>	a number in standard scientific notation	4.556000e+01
<code>%s</code>	a string	"Java is cool"

Here is an example:

*****Same as the unnumbered figure in intro6e p65**

```
int count = 5;
double amount = 45.56;
printf("count is %d and amount is %f", count, amount);
```

display count is 5 and amount is 45.560000

Items must match the specifiers in order, in number, and in exact type. For example, the specifier for `count` is `%d` and for `amount` is `%f`. By default, a floating-point value is displayed with six digits after the decimal point. You can specify the width and precision in a specifier, as shown in the examples in Table 2.

Table 2
Examples of Specifying Width and Precision

Example	Output
<u>%5c</u>	Output the character and add four spaces before the character item.
<u>%6b</u>	Output the boolean value and add one space before the false value and two spaces before the true value.
<u>%5d</u>	Output the integer item with width at least 5. If the number of digits in the item is < 5, add spaces before the number. If the number of digits in the item is > 5, the width is automatically increased.
<u>%10.2f</u>	Output the floating-point item with width at least 10 including a decimal point and two digits after the point. Thus there are 7 digits allocated before the decimal point. If the number of digits before the decimal in the item is < 7, add spaces before the number. If the number of digits before the decimal in the item is > 7, the width is automatically increased.
<u>%10.2e</u>	Output the floating-point item with width at least 10 including a decimal point, two digits after the point and the exponent part. If the displayed number in scientific notation has width less than 10, add spaces before the number.
<u>%12s</u>	Output the string with width at least 12 characters. If the string item has less than 12 characters, add spaces before the string. If the string item has more than 12 characters, the width is automatically increased.

<Side Remark: left justify>

You can put the minus sign (-) in the specifier to specify that the item is left-justified in the output within the specified field. For example, the following statement

```
printf("%8d%-8s\n", 1234, "Java");
printf("%-8d%-8s\n", 1234, "Java");
```

displays

```
      1234Java
1234      Java
```

CAUTION: The items must match the specifiers in exact type. The item for the specifier %f or %e must be a floating-point type value such as 40.0, not 40. Thus an int variable cannot match %f or %e.

TIP: The % sign denotes a specifier. To output a literal % in the format string, use %%.