

# Chapter 3 – Exploring Cout

## 1. New Lines

The object `cout` outputs the results of expressions. Expressions can contain information that effects how the output appears. One of the most useful is `endl`, which makes output change lines. For example:

```
cout << "This is on one line." << endl << "This is on another line.";
```

will produce the output

```
This is on one line.  
This is on another line.
```

Note: `cout` and `endl` come from `iostream`.

## 2. Backslash Commands

When used in a character or string expression, the backslash character (`\`) is used to give instructions to `cout`. For example, `\n` has the identical effect of `endl`. For example:

```
cout << "This is on one line." << "\n" << "This is on another line.";
```

will produce the output

```
This is on one line.  
This is on another line
```

The only advantage of `\n` is that it can be included in a string. For example:

```
cout << "This is on one line.\nThis is on another line.";
```

will also produce the output

```
This is on one line.  
This is on another line
```

Backslash commands originated with the C programming language. The following is a partial list.

### **Partial Table of Backslash Commands to Cout**

<i>Character</i>	<i>Outputs</i>
'\n'	new line characters
'\b'	backspace one characters
'\f'	form feed
'\r'	carriage return
'\t'	horizontal tab
'\v'	vertical tab
'\\'	backslash
'\''	single quote
'\"'	double quote

## **3. Member Functions**

*Member functions* are functions in a *class*. Each *object* of a *class* will have as part of itself the member functions of its class. Member functions may be invoked (called, used) in the following form:

*object.function ( argument-list )*

The object *cout* has member functions. Two commonly used *cout* member functions are *fill* and *width*.

*cout* member function *width* sets the spacing of output. *Width* is used as in the following example,

```
cout << "Hi!";
cout.width(10);
cout << "Bye!";
```

would produce the following output:

*Hi!      Bye!*

The expression "Bye!" takes up 10 spaces. When it is output, the expression produces 6 spaces and B-y-e-!.

The effect of *width* is limited to the next *cout* statement. For example,

```
cout << "Hi!";
cout.width(10);
cout << "Bye!";
cout << "Why?"
```

would produce the following output:

```
Hi!    Bye!Why?
```

*cout* member function *fill* sets the character that will be used in output that has a width larger than the expression needs. For example,

```
cout.fill('#');
cout << "Hi!";
cout.width(10);
cout << "Bye!";
cout.width(6);
cout << "Why?"
cout.width(4);
cout << 88;
```

would produce the following output:

```
Hi!#####Bye!##Why?##88
```

Note that the effect of *fill* continues beyond the first *cout*.

## 4. Decimals

The number of decimal places output from the results of a floating point number expression can be controlled with the *ostream member functions setf* and *cout* member function *precision*. In the following example, the *setf* function is called to enable fixing the location of the decimal point. The *precision* function, *cout* is set to a *precision* of 1. This will produce a number with a one digit decimal.

```
double d = 44.1234;
cout.setf(ios::fixed, ios::floatfield);           // include these setf calls
cout.setf(ios::showpoint);                       // once per program
cout.precision(1);
cout << d;
```

The output of this example code is:

```
44.1
```

As usual, cout will round any number output, as in this example:

```
double d = 44.8765;
cout.setf(ios::fixed, ios::floatfield);
cout.setf(ios::showpoint);
cout.precision(2);
cout << d;
```

The output of this example code is:

44.88

*Note: The default setting of cout precision is 6.*

In addition to *cout.width()* and *cout.precision()*, there are functions that do not belong to object *cout* and can be placed directly in a cout statement as an expression. These are functions *setw* (sets the width of the next output) and *setprecision* (sets the precision of the next floating point output), which can be used in a C++ program by including *iomanip*. Here is an example program using these functions.

```
#include <iostream>
#include <iomanip>

using namespace std;

void main( )
{
    double d = 44.8765;
    cout.setf(ios::fixed, ios::floatfield);
    cout.setf(ios::showpoint);
    cout.fill('*');
    cout << "d is " << setw(10) << setprecision(2) << d << endl;
}
```

This program will produce the output:

*d is \*\*\*\*\*44.88*

## **Exercises**

1. What is output of the following *cout* statement?

```
cout << "one" << endl << "two\nthree\b\b\b\bfour";
```

2. What is the output of the following statements?

```
cout.fill('!');  
cout.width(8);  
cout << "one";  
cout.width(6);  
cout << 4 * 2;
```

3. What is the output of the following statements?

```
double PI = 3.141592654;  
cout.setf(ios::fixed, ios::floatfield);  
cout.setf(ios::showpoint);  
cout << PI << endl;  
cout.precision(0);  
cout << PI << endl;  
cout.precision(1);  
cout << PI << endl;  
cout.precision(2);  
cout << PI << endl;  
cout.precision(4);  
cout << PI << endl;  
cout.precision(8);  
cout << PI << endl;  
cout.precision(9);  
cout << PI << endl;  
cout.precision(10);  
cout << PI << endl;  
cout.precision(5);  
cout << PI << endl;
```

4. Rewrite the code of exercise 3, replacing the calls to *cout.precision* with calls to *setprecision*.

**Programming Assignment 3.1**

Write a the simplest program possible to output the following:

```

CCC  +    +
C    +    +
C    +++++ +++++
C    +    +
CCC  +    +

```

**Programming Assignment 3.2**

Write a program to list the following names in 3 rows and 3 columns. Hint: Use ‘\t’ to make the columns.

Abe, Bill, Carrie, David, Edward, Gale, Harry, Ida, Jenny

**Programming Assignment 3.3**

Rewrite programming assignment 3.2 so that the spaces between the three names on one row are replaced by the ampersand character (&). Note: You must use the *fill* function.

**Programming Assignment 3.4**

Write a program to output the area of a circle that is 8.375 meters in diameter. (Use the formula  $3.141592654 * \text{radius} * \text{radius}$ ). Limit the number of decimal digits displayed to 2 decimal places.