Manipulator	Function	Example	Output
endl	Prints a newline and flushes output buffer	*cout* **<< 5 <<** *endl* **<< 10 <<***endl***;**	5 10
setw()	Assigns the width of the next item to be displayed	*cout* **<<** *setw***(4) << 5;** *cout* **<<** *setw***(4) <<10;**	5 10
setprecision()	Assigns the number of fractional digits printed.	*cout* **<<** *fixed* **<<** *setprecision***(1) << 3.14159;**	3.1
scientific	Prints floating-point numbers in scientific format.	*cout* **<<** *scientific* **<< 3.14;**	3.14E+000
fixed	Prints floating-point numbers in fixed format	*cout* **<<** *fixed* **<< 1.61e1;**	16.1
showpoint	Prints the decimal point for floating–point numbers even if there are no decimal digits.	*cout* **<<** *showpoint* **<< 1.6e1;**	16.
left	Left-justifies output	*cout* **<<** *setw***(4) <<** *left* **<<16;**	16
right	Right-justifies output	*cout* **<<** *setw***(4) <<** *right* **<< 16;**	16
dec	Assigns integer base to be base 10	*cout << dec <<* **16***;*	16
oct	Assigns integer base to be base 8	*cout << oct <<* **16***;*	20
hex	Assigns integer base to be base 16	*cout << hex <<* **16***;*	10
uppercase	Coerce the exponent e to E	*cout* **<<** *scientific* **<<** *setprecision***(3) <<** *uppercase* **<< 1.602e-19;**	1.602E-19
nouppercase	Coerce the exponent E to e	*cout* **<<** *scientific* **<<** *setprecision***(3) <<** *nouppercase* **<< 1.602E-19;**	1.602e-19
boolalpha	Prints **true** as true	*cout << boolapha <<* **true**	true
noboolalpha	Prints **true** as 1	*cout << noboolalpha <<* **true**	1
setfill()	Assigns the padding character	*cout << setw***(5)** *<< setfill***('0')** **<< 34;**	00034

Table 1. Manipulators

#include <*iomanip***>**

Examples:

```
include <iostream>
#include <iomanip>
using namespace std;
int main()
{    int i(29);
     cout<<endl<<"setw( 5)<<setfill('0')<< left=  "<<"|"<<setw( 5)<<setfill('0')<<left<<i<<"|";
     cout<<endl<<"setw( 5)<<setfill('0')<<right= "<<"|"<<setw( 5)<<setfill('0')<<right<<i<< "|";

     cout<<endl<<"setw(10)<<setfill(' ')<< left=  "<<"|"<<setw(10)<<setfill(' ')<<left<<i<<"|";
     cout<<endl<<"setw(10)<<setfill(' ')<<right= "<<"|"<<setw(10)<<setfill(' ')<<right<<i<< "|";

     cout<<endl<<"setw( 1)<<setfill(' ')<< left=  "<<"|"<<setw( 1)<<setfill(' ')<<left<<i<<"|";
     cout<<endl<<"setw( 1)<<setfill(' ')<<right= "<<"|"<<setw( 1)<<setfill(' ')<<right<<i<< "|";

     cout<<endl;
     return 0;
}
```

Figure 1. Program **p01**, *endl*, *setw()*, *right* and *left* manipulators.

Program **p01** prints:

```
setw( 5) << setfill('0') << left=  |29000|
setw( 5) << setfill('0') << right= |00029|
setw(10) << setfill(' ') << left=  |29        |
setw(10) << setfill(' ') << right= |        29|
setw( 1) << setfill(' ') << left=  |29|
setw( 1) << setfill(' ') << right= |29|
```

Program **p01** notes:

1. Vertical bars | delimit the formatted output.
2. The manipulator *setw***()** specifies the field width of the output provided that the field width is greater than or equal to the actual output width required. For example, integer variable *i*, having the value 29, requires a field width of two characters. Specifying a field width of a single character using the *setw***()** manipulator has no effect on the minimum width of the output field.
3. The manipulator setfill() specifies the character used to fill the spaces unfilled by the actual output. For example if the output field width is 5 (setw(5)) and the actual output (29) requires two characters and the fill-character

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{   double pi=3.14159265;
    cout<<endl<<"fixed<<setprecision(0) pi="<< fixed<<setprecision(0)<<pi;
    cout<<endl<<"fixed<<setprecision(1) pi="<< fixed<<setprecision(1)<<pi;
    cout<<endl<<"fixed<<setprecision(2) pi="<< fixed<<setprecision(2)<<pi;
    cout<<endl<<"fixed<<setprecision(3) pi="<< fixed<<setprecision(3)<<pi;
    cout<<endl<<"fixed<<setprecision(4) pi="<< fixed<<setprecision(4)<<pi;
    cout<<endl<<"fixed<<setprecision(5) pi="<< fixed<<setprecision(5)<<pi;
    cout<<endl;
    return 0;
}
```

Figure 2. Program **p02**, *fixed* and *setprecision***()** manipulators.

Program **p02** prints:
fixed<<setprecision(0) pi=3
fixed<<setprecision(1) pi=3.1
fixed<<setprecision(2) pi=3.14
fixed<<setprecision(3) pi=3.142
fixed<<setprecision(4) pi=3.1416
fixed<<setprecision(5) pi=3.14159

Program **p02** notes:

1. The *fixed* manipulator determines the format for a floating-point value. The *fixed* manipulator specifies that a floating-point value is formatted as an integer to the left of the decimal point and a fractional value to the right of the decimal point. No exponent is printed.