Reading/Writing One Character at a Time

- get (char&)   read single character from stream
- put (char)   write single character to stream
- int eof( )   whether the next character is EOF

```
char  next;
int  blank_count = 0;

do  {

        cin.get (next);
        if   ( cin.eof( ) )
            break;

        if ( next ==  '   ' )
            blank_count ++ ;
        else
            cout.put  ( next ) ;

}   while  ( 1 );

cout  << "Number of blanks  =  "
            <<  blank_count  << endl;
```

Reading a number as a string of characters

```
// Reads consecutive characters and converts to int
#include <iostream.h>
#include <ctype.h>

void read_int (int& integer_value) {
        const int base = 10;
        const char nwln = '\n';

        char next;
        int digit;

        cout << "Enter on line of characters,"
                << " following by a carrage return." << end;

        integer_value = 0;
        cin.get (next);

        while (next != nwln) {
                if ( isdigit (next) ) {
                   digit = int(next) - int('0');
                   integer_value = base * integer_value + digit;
                }
                cin.get (next;
        }
}
```

Stream I/O manipulators

Stream (of type ios) are complex data objects

- As an infinite sequence of bytes/characters
- Has designated operators (>> , <<)
- Has designated functions (get, put, etc.)
- Has modification settings (flags) to change the I/O behavior
- Has manipulators and function to modify the flags

Input extraction:

- Whether whitespace is significant

Output formatting:

- Field width - number of spaces used to display a data item
- Adjustment - to the left or the right
- Precision - in what precision (for real numbers)

I/O Flags

Name	Meaning if on	Default
Input:		
skipws	Skip whitespace on input	On
Output adjustment:		
left	Left adjust output in field	Off
right	Right adjust output in field	Off
Output floating number format:		
fixed	Use fixed point format (e.g. 59.06)	Off
scientific	Use scientific format (e.g. 5.906E1)	Off
showpoint	Show decimal point and trailing zeros (e.g. 59.00)	Off

I/O Manipulators and Functions

C++ I/O manipulator library:

#include < iomanip.h >

Manipulator	Function	Description
setioflags	setf	Turn on a format flag
resetiosflags	unsetf	Turn off a format flag
setprecision	precision	Set precision of output numbers
setw	width	Set the field width for next output

Examples

```
double x = 32.1630,  y = 67009.1223,  z = 1000.84856 ;

// Print all three numbers in fixed notation with precision 1
cout    << setioflags ( ios :: fixed )  << setprecision (1)
        << "Fixed notation, precision = 1\n"
        << "x = " << x << "\ny = " << y << "\nz = " << z << endl;

// Print all three numbers in scientific notation with precision 3
cout    << resetioflags( ios :: fixed ) << setiosflags( ios :: scientific )
        << setprecision(3) << "\nScientific notation, precision = 3\n"
        << "x = " << x << "\ny = " << y << "\nz = " << z << endl;

// Print x fixed, y scientific, z fixed
cout    << "\nPrecision = 3, x is fixed, y scientific, z fixed\n"
        << resetioflags( ios :: scientific ) << setiosflags( ios :: fixed )
        << "x = " << x
        << resetioflags( ios :: fixed ) << setiosflags( ios :: scientific )
        << "\ny = " << y
        << resetioflags( ios :: scientific ) << setiosflags( ios :: fixed )
        << "\nz = " << z << endl;
```

Output:

```
Fixed notation, precision = 1
x = 32.2
y = 67009.1
z = 1000.8

Scientific notation, precision = 3
x = 3.216e+01
y = 6.701e+04
z = 1.001e+03

Precision = 3, x is fixed, y scientific, z fixed
x = 32.163
y = 6.701e+04
z = 1000.849
```

Examples

```
double price = 127.1;
double taxRate = 0.065

cout    << setiosflags( ios :: fixed << setprecision (2)
        << "Price     $" << set(7) << price
        << "\nT  ax     $" << setw(7) << price * taxRate << " at "
        << setprecision (1) << taxRate * 100 << "%\n"
        << setprecision (2) << "T    otal     $"
        << setw(7) << price * (1 + taxRate) << endl ;
```

Output:

```
Price     $ 127.01
Tax       $    8.31 at 6.5%
Total     $ 135.36
```

```
cout   << "12345678901234567890\n";
cout   << setw(10)  << "Hello"  << endl;
cout   << "Hello"  << endl;
cout   << setw(1) << "Hello" << setw(10 << "Hello" << endl;
```

Output:

```
12345678901234567890
```