

Objectives: You will gain experience using C++:

- assignment statements
- C++ arithmetic expressions
- cmath library
- rand() and srand() functions from the cstdlib library
- File input and output

Download the following file to your desktop: <http://www.cs.uni.edu/~fienup/cs051f09/labs/lab2.zip>
Extract this file by right-clicking on lab2.zip icon and selecting Extract All.

Part A: Multiple Assignment Statements -- the = can be used to assign a value to multiple variables:

```
x = y = z = 5;
```

The = operator is also a binary operator that associates right to left: $x = (y = (z = 5))$. The value of the assignment operation = is the value that is assigned, so $(z = 5)$ has the value 5 which gets assigned to y, etc.

Start Visual Studio with the testNumbers project by double-clicking on the lab2/testNumbers/testNumbers.sln file. When you run the program using Debug | Start Debugging, you should see the following console window:

```
Assignment: "myDouble = myInt = myOtherDouble = 3.9;"
myDouble: 3 myInt: 3 myOtherDouble: 3.9

Assignment: "myDouble = myOtherDouble = myInt = 3.9;"
myDouble: 3 myOtherDouble: 3 myInt: 3

Assignment: "myDouble = myInt = myOtherInt = 7;"
Assignment: "myInt += 3;"
Assignment: "myOtherInt /= 3;"
Assignment: "myDouble /= 3;"
myInt: 10 myOtherDouble: 3 myDouble: 2.33333
Hit any key to continue...
```

a) Explain what happens with the assignment statement: `"myDouble = myInt = myOtherDouble = 3.9;"`

b) Explain what happens with the assignment statement: `"myDouble = myOtherDouble = myInt = 3.9;"`

c) Explain the last line of the output: `"myInt: 10 myOtherDouble: 3 myDouble: 2.33333"`

When you are satisfied with your answers (or hopelessly stuck), raise your hand and have an instructor check your answers.

Part B: The header file `iomanip` contains the following manipulators to control the formatting of the output (and input) of numbers and strings.

Stream Manipulator	Description
<code>setw(<i>n</i>)</code>	Establishes a print field of at least size <i>n</i>
<code>fixed</code>	Displays floating-point numbers in fixed-point notation
<code>scientific</code>	Displays floating-point numbers in scientific notation
<code>showpoint</code>	Causes a decimal point and trailing zeroes to be displayed, even if there is no fractional part.
<code>setprecision(<i>n</i>)</code>	Sets the precision of floating-point numbers to <i>n</i>
<code>left</code>	Left justify the output
<code>right</code>	Right justify the output

Start Visual Studio with the `Test_iomanip` project by double-clicking on the `lab2/Test_iomanip/Test_iomanip.sln` file. When you run the program using Debug | Start Debugging, you should see the following console window:

```
(default) with << setw(15) << right:
setprecision(3) << 1234.5678:      1.23e+003
setprecision(4) << 1234.5678:      1235
setprecision(6) << 1234.5678:      1234.57

cout << fixed
setprecision(3) << 1234.5678:      1234.568
setprecision(4) << 1234.5678:      1234.5678
setprecision(6) << 1234.5678:      1234.567800

cout << scientific
setprecision(3) << 1234.5678:      1.235e+003
setprecision(4) << 1234.5678:      1.2346e+003
setprecision(6) << 1234.5678:      1.234568e+003
setprecision(6) << "1234.5678":      1234.5678
```

- a) Explain the meaning of `setprecision` in the default notation.

- b) Explain the meaning of `setprecision` in the fixed notation.

- c) Explain the meaning of `setprecision` in the scientific notation.

- d) The `setw` manipulator can also be used on input of a character string to avoid buffer overrun. The `cin` applies the `setw` field width to the very next item entered by the user, and `cin` stops reading when it encounters a whitespace character (space, tab, or newline/<Enter> character).

The cin object also has the following formatted input methods.

Example	Description
<code>cin.getline(charStringVar, SIZE);</code>	Reads a whole line into the charStringVar or until (SIZE-1) characters has been read (the last spot in the array is saved for the NULL character)
<code>cin.get(chVar);</code>	Reads the next character regardless of what it is.
<code>cin.ignore();</code>	Reads the next character and throws it away.
<code>cin.ignore(20, '\n');</code>	Discards the next 20 characters of input or until the '\n' character is read. In general the first argument can be any integer and the second can be any character.

Start Visual Studio with the TestStringsRevisited project by double-clicking on the lab2/TestStringsRevisited/TestStringsRevisited.sln file. Several runs of this program are shown below.

Program	Window's Visual Studio Output
<pre>#include <iostream> #include <iomanip> using namespace std; int main() { const int SIZE = 11; char str1[SIZE]; char str2[SIZE]; cout << "Enter a string: "; cin >> setw(SIZE) >> str1; cout << "Enter another string: "; cin.getline(str2,SIZE); cout << "str1 is \"\" << str1 << \"\" and str2 is \"\" << str2 << \"\" << endl; char ch; cout << endl << "Hit any key to continue...." << endl; cin.get(ch); return 0; } // end main</pre>	<pre>Enter a string: abcdefghijklmn Enter another string: str1 is "abcdefghij" and str2 is "klmn" Hit any key to continue.... Enter a string: Mark Enter another string: str1 is "Mark" and str2 is "" Hit any key to continue.... Enter a string: Mark Fienup Enter another string: str1 is "Mark" and str2 is " Fienup" Hit any key to continue....</pre>

For each of the above examples, explain why the user was only able to enter the first string.

e) What would we need to do to fix the above program?

When you are satisfied with your answers (or hopelessly stuck), raise your hand and have an instructor check your answers.

Part C: Useful library functions. The following table contains some useful library functions:

Header File	Function Example	Description
cmath	<code>y = abs(x)</code>	Returns the absolute value of an integer argument
	<code>y = fmod(x, z)</code>	Returns the remainder of x divided by z where both arguments and the returned value are doubles
	<code>y = log(x)</code>	Returns the natural logarithm of the argument. The argument and the return type are doubles.
	<code>y = log10(x)</code>	Returns the base-10 logarithm of the argument. The argument and the return type are doubles.
	<code>y = sqrt(x)</code>	Returns the square root of the argument. The argument and the return type are doubles.
	<code>y = pow(x, z)</code>	Returns the first argument raised to the second argument power.
cstdlib	<code>y = rand()</code>	Returns the a pseudorandom number as an int.
	<code>srand(seed)</code>	The unsigned int argument acts as a seed value to the pseudorandom number generator.
ctime	<code>seed = time(0);</code>	Returns the number of seconds elapsed since midnight, Jan. 1, 1970. (The result can be used as the seed to srand)

For example, the following C++ program (project folder PythagoreanTheorem) calculates the hypotenuse of a right triangle using the Pythagorean theorem: $c = \sqrt{a^2 + b^2}$, where c is the length of the hypotenuse, and a and b are the other sides of the triangle. The sides of the triangle (a and b) are generated at random such that a is between 1 and 10, and b is between 5 and 25.

```
#include <iostream>
#include <cmath>
#include <cstdlib>
#include <ctime>
using namespace std;

int main() {
    const int MAX_RANGE_OF_SIDE_A = 10;
    double a, b, c;
    unsigned seed;

    // Seed the random number generator
    seed = time(0);           // Get the system time
    srand(seed);

    // Randomly generate the sides a and b
    a = rand() % 10 + 1;     // random value between 1 and 10
    b = rand() % 21 + 5;    // random value between 5 and 25
    cout << "The length of side a is " << a
         << " and the length of side b is " << b << endl;

    // Calculate the hypotenuse using Pythagorean Theorem
    c = sqrt(pow(a, 2.0) + pow(b, 2.0));

    cout << "The length of the hypotenuse is " << c << endl;

    char ch;
    cout << "\n Hit any key to exit..." << endl;
    cin.get(ch);
    return 0;
} // end main
```

Write a program that interactively takes two integers (say 50 and 100) and randomly generates a number between them. **Print your program file (.cpp) to turn in -- make sure you include your name.**

When you are satisfied with your answers (or hopelessly stuck), raise your hand and have an instructor check your answers. Do not forget to "log off" of the machine you are working on.