Functions and libraries

- we've seen a few examples of the use of libraries (cstdio, iostream, iomanip) as collections of pre-written code
- each library can define a variety of things: data types, constants, functions, etc
- for “built-in” libraries we use the #include <libraryname> syntax to allow our program to use the items defined in the library
- there are many such libraries: cstdio, iostream, cmath, cstring, cctype, cstdlib, string, climits, and cfloat are a few we'll use
- later we'll introduce syntax for creating and using our own libraries of code

Functions

- a function is simply a named collection of code that we can run within our programs
- printf and scanf are examples of functions we've already used
- to run, or *call*, a function we invoke its name and *pass* any data values it requires, these data values are referred to as *parameters* or *arguments*
- for the call below, “%f” and x are the two parameters passed

```
printf(“%f”, x);
```

Return values

- in addition to whatever computations they may perform or actions they may take, functions can also *return* a value
- once the function completes, it sends a value back to the spot where it was called, where the program can use it
- the sqrt function (from the cmath library) computes the square root of its parameter and returns it

```
x = sqrt(y);
```

- in the example above, sqrt computes the root of y and then returns that value, which we then store in x

Using functions

- to use a function effectively, we need to know:
  - its name
  - which library it is declared in
  - the types of parameters it expects, in what order
  - the type of data value (if any) it returns when complete
- when a function is called, the caller (e.g. the main routine) waits for the function to complete before resuming

printf and scanf return values

- printf and scanf each return integer values (in addition to their other actions)
- printf returns a count of how many characters it displayed

```
int x = 3;
int count;
count = printf("x is %d\n", x);
// displays "x is 3" and a newline
// count will thus be assigned 7
```

- scanf returns a count of how many variables it successfully stored a value in (we'll use this for error checking later)

Some cmath functions

- the cmath library includes many common math functions, including:
  - sqrt(x) - returns the square root of x
  - pow(x,y) - returns x to the power of y
  - fabs(x) - returns the absolute value of x
  - ceil(x) - returns x rounded up
  - floor(x) - returns x rounded down
  - cos(x) - returns the cosine of x
  - plus sin(x), tan(x), log(x), and many others
- (none of these actually change x or y)

Some cctype functions

- these functions are used to work with chars
  - toupper(x) - return the uppercase char for x
  - tolower(x) - return the lowercase
  - isalpha(x) - check if x contains an alphabetic character
  - isspace(x) - check if x contains a whitespace character
  - plus isupper(x), islower(x), ispunct(x), isdigit(x), etc
- again, none of these actually alter the character passed to them, they compute and return a new result