Functions

- A *function* is a set of instructions
  - When *executed*, it accomplishes a task
- Most functions require *input parameters*
  - Pieces of data that the function needs to do its job
  - This is **not the same** as input from the user
- Many functions *output* a *return value*
  - A piece of data that is the result of that job
  - This is **not the same** as output to the screen

Calling Functions

- Every function has 3 parts you need to know in order to use it (besides what it does, of course):
  - A name
    - Follows the same rules as variables names
    - Can't be the same as the name of a variable or a reserved word
  - A *parameter list*
    - These are the input values that the function needs in order to do its job
    - Each parameter is expected to be a specific data type (int, double, char, string, etc)
  - A *return type*
    - This is the data type that the function returns when it is done

Calling Functions

- Functions are *called* by name:

```
y = math.cos( x )
```

- When you call a function, you have to provide it with appropriate parameter values
  - The number of values it expects…
  - …in the order it expects them

- We say these values are *passed in* to the function

- Paramters are also called *arguments*

Using the Return Value

- Many functions *return* a value
  - Sometimes we call this the *output* of the function
    - This is different than output to the screen!
  - Return values aren’t printed to the screen, they are returned to the calling statement
    - A function call expression evaluates to its return value
- Some examples:

`x = 3 + 4`

evaluates to: `x = 7`

`y = math.sqrt( 16.0 )`

evaluates to: `y = 4`

Function Calls and Return Values

- When calling a function, you typically:
  - Save the return value for further calculation
  - Use the return value in some calculation
  - Print the return value

- In other words, functions are called:
  - In an assignment statement
  - In an expression
  - As an actual parameter to another function