CSE 2010:
Week 2

Chapter 2: Using Pre-Defined C++ Libraries and their Functions

What this lecture covers:

- The cmath pre-defined C++ library
- Using cmath functions

Pre-defined C++ Libraries

- Background:
  - C++ library includes several pre-defined libraries that contain function that you can use in your code.
  - You have access to these libraries and can use them in your program with the #include statement

    ```
    #include <libraryName>
    ```
  - You can see a list of some here: http://www.cplusplus.com/reference/

Using Pre-defined Functions

- Your main function (`int main()`) is what controls the C++ program, but within main, you can call other functions to complete specific calculations or tasks.
- Definition of Functions:
  - A function is a group of statements that perform a specific tasks.
- By including pre-defined libraries, we have access to the functions within them. We cannot see the code for these functions, but we can still use them.
- Syntax for using predefined functions:
  - `functionName(parameters)`
    - Each function has a unique name
    - Each function requires and accepts specific parameters. Parameters are the values that the function needs in order to return a specific value to you.
    - Each function returns up to a single value.
    - The number of parameters can vary on the function. Some have none, others have 2 or 3, etc.

<cmath> - The C numerics library

http://www.cplusplus.com/reference/cmath/

- To use functions in cmath, have the following line in your program, right after `#include <iostream>`:
  - `#include <cmath>`
- Functions to compute trigonometry related calculations:
  - `cos(x) //returns the cosine of the value x`
  - `sin(x) //returns the sine of the value x`
  - `tan(x) //returns the tangent of the value x`
- Exponential and logarithmic functions
  - `log(x) //returns the log of x`
  - `log2(x) // returns the log base 2 of x`
- Power functions:
  - `pow(x,n) // returns the value x raised to the n`
  - `sqrt(x) //returns the square root of x`
- You can use these functions anywhere that you would use a regular value
  - cout statement
  - variable definition
  - arithmetic expressions
  - as the parameter for another function call

Program example of using pow(), where the function call is used in two different ways, but we would get the same output

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

int main() {
    //function call as part of a cout statement
    int a = 7, b = 3;
    cout << a << “ raised to the power of “ << b << “ is: “ << pow(a,b) <<“\n”;

    //function call as part of a variable definition
    int value = pow(a,b);
    cout << a << “ raised to the power of “ << b << “ is: “ << value << “\n”;
    return 0;
}
```

Output of program above:

```
7 raised to the power of 3 is 343

7 raised to the power of 3 is 343
```

Things to remember when converting mathematical expressions to valid C++ expressions

- You are limited to characters that are on your keyboard
  - No square root symbol
  - No log
  - No exponents or subscripts
- Before you implement the statement in your program, make sure you have all the necessary variables/values
  - If your formula/expression uses specific variables, be sure that they have been declared.
- Include all necessary libraries
- Let’s try an example with the quadratic formula!