# Pre-Lab for Lab 2

## Lab Overview

Brief description of the purpose of this lab

## Pre-Lab Submission

Complete this Pre-Lab and submit it as a PDF by 23:59 the night before your lab section for Lab XX.

* Complete the following sections of this document before submitting:
	+ Lab Prep 1: Declaring a Constant
	+ Lab Prep 2: Use math functions to compute the length of a cone
	+ Lab Prep 3: Format floating point output

## Learning Goals

List of learning goals:

* Declaring Constants
* Using math functions to compute values
* Formatting output

### Goal 1: Declaring Constants

Tips for declaring constants:

* use uppercase for constants
* put constants near the top of the file
	+ after library includes
	+ before the main routine
* the const keyword makes this a constant, so we can only set its value once. Like PI, it will never change.
* To do this we will need to define a constant to hold the value PI:
* const double PI = 3.14;

#### Exemplars

##### Declaring a constant

Figure 2: Example of declaring a constant for a tax rate. Note the const key word which indicates that this is a constant, type matches what was needed, and the name is declared as ALL CAPS.

#include <cstdio>

const float GST = 0.05;//because it is 5% in BC

int main(){

...

Figure 1: Description of the data we need to store in the constant to represent a tax rate

Suppose we need a constant to hold the GST tax rate:

* Its type will be a floating point number, because the value is < 1 and > 0
* It can be named GST

Lab Prep 1: Declaring a Constant

Complete the given examples by filling in the box with your solution. In each case, carefully read the description and complete the code so that it matches the description. Refer to the exemplars provided above for hints about how to proceed.

##### Declare a constant

Declare a constant to hold the value of Pi to 5 decimal places

Your answer:

### Goal 2: Using Math Functions

**Including the math library:**

Before we can use math functions, we need to *include* the math library. *Include* statements are typically placed at the top of a C++ file, after the header comment. To *include* the math library, add the following line to your file:

#include <cmath>

**Computing exponents in C++.**

Exponents are computed in C++ using the function *pow*:

* the function pow takes 2 parameters:
	+ base (in this case the value in length)
	+ exponent (in this case, 3.0)
* These parameters are expected to be of type double.
	+ when we pass in the int length, it gets converted to a double. Is information lost?
* pow will *return* a double, as well. What happens when we *assign* it to a variable that is of type int?
* More information is available here: http://www.cplusplus.com/reference/cmath/pow/

#### Exemplars

##### Using the pow function

Figure 4: C++ version of above

double base = 10.0;

double exp = 4.0;

double answer = pow(base, exp)

Figure 3: Formula that requires the use of the pow function

Compute 104

**Computing the square root of a number in C++:**

The square root of a number can be computed in C++ using the function *sqrt*:

* the function sqrt takes 1 parameter:
	+ the value we want the square root of
* This parameter is expected to be of type double.
* sqrt will *return* a double.
* More information is available here: http://www.cplusplus.com/reference/cmath/sqrt/

##### Using the sqrt function

Figure 6: Using the sqrt function and saving the result. Both the parameter (area) and the return value will be doubles.

double squareRoot = sqrt(area);

Figure 5: Using the sqrt function

Find the square root of a value stored in the variable area.

#### Student Practice 2

Complete the given examples by filling in the boxes with your solution. In each case, carefully read the description and complete the code so that it matches the description. Refer to the exemplars provided above for hints about how to proceed.

Lab Prep 2: Use math functions to compute the length (l) of a cone



l

The length (l) of a cone is computed as:

* + Length = √ (radius2 + height2)

Use the pow function to rewrite this as a C++ expression, below:

Your answer:

### Goal 3: Formatting output using printf

We can get printf to format the output of our numbers so that it looks better. This is especially useful for $$ output.

To tell printf how to format the floating point value that is passed in, we put a format symbol between the '%' and the 'f'.

Here .3 means "three digits to the right of the .".

You can find more information on printf and formatting, here: http://www.cplusplus.com/reference/cstdio/printf/

When we output the values for the sphere, we want to format the output so that there are exactly 3 values after the decimal.

#### Exemplars

##### Formatting a floating point number

*Figure 7: A description of what formatting is desired*

float myNum = 6.1234567;

printf("my num is %.4f", myNum);

//this will output: my num is 6.1234

Format a floating point number using printf so there are exactly 4 values to the right of the decimal:

*Figure 8: Specifying that exactly 4 values should be shown to the right of the decimal. Note that the string ‘.4’ is put between the % and f that normally indicate that the input number is a float.*

#### Student Practice 3

Complete the given examples by filling in the boxes with your solution. In each case, carefully read the description and complete the code so that it matches the description. Refer to the exemplars provided above for hints about how to proceed.

Lab Prep 3: Format output

Declare a variable to hold the value 9.99999.

Use printf to output it with:

1. 2 values to the right of the decimal
2. 0 values to the right of the decimal

Your answer: