# Using the Built-In Functions of the Math Library

**The goal for this exercise** is to make sure that you can use the built-in Math functions. You are not required to understand the mathematics itself (i.e., you need to know how to call Sine & Cosine, but it’s ok if you don’t know/remember trigonometry).

**What you need to do for this exercise:**  In the starter project, fill in the Built\_In\_Math\_Functions.RunExercise method, so that it will call each of the following built-in functions from:

* Pow() // ask the user for a base, and an exponent
 // and raise the base to that exponent
* Sqrt() // Ask the user for a number to take
 // the square root of
* Sin() // Print out the values of sin(x)
 // where x = 0...2Pi, increasing x by
 // .1Pi **(**i.e., print sin(0), sin(0.1\* Pi),
 // sin(0.2\* Pi), ... , sin(2\*Pi) **)**
* Cos() // Print out the values of cosine from 0...2Pi, // by .1Pi (i.e., 0, 0.1\* Pi, 0.2\* Pi,..., 2\*Pi
 // Note that you should use Math.PI rather than
 // making your own "Pi".

# You should write your code so that you don't need 20 separate lines of code, each of which does a single call to Sin/Cos in your code ☺

# A class (in object-oriented terms) is a overall description of how all individual examples of a particular class of software (or category of software, or type, or whatever you want to call it) are the same – for example, a Car class might describe what all cars have in common (weight, maximum rate of acceleration, et). An instance of a class is one particular, individual, unique object that belongs to that class/category/type – for example, your own, personal car might have a weight of 2,000 pounds, and accelerate at the rate of 10 mph, per second.

# In the case of the above math routines, there is no instance to use – instead, you use the class itself to get to the methods. Another example of this is the Int32.Parse method/command – in both of these situations, we want to use the various methods, but we don't necessarily want to create an instance of the Math class (or the Int32 class) just to call a single method (especially since we wouldn't use that instance again later). You'll see this again in the next exercise, when we create a method (main) that must be connected to the overall class, since it can't be connected to a specific instance of the class.

# You'll notice that all methods (aka functions, subroutines, procedures, etc, more-or-less) must be called either on an existing class, or a specific instance of a class.