**Program #1 –Working with Mathematical functions**

Computers were originally about doing mathematical work. Lots of mathematical work. And you can do this in C# using the Math library (static) functions..

As we saw in our Chapter 3 Lecture PowerPoint, we can use Math library for many mathematical functions.

We will start by looking at three trigonometric functions: sin, cos, and tan shown below.

**x = Math.Sin(rads)**

**y = Math.Cos(rads)**

**z = Math.Tan(rads)**

Please keep in mind that the argument to each function is in radians, ***NOT*** degrees. You can convert degrees into radians by multiplying by M**ath.PI** and dividing by 180.

Write three of your own functions, **mysin()**, **mycos()**, **mytan()**, that will accept an angle in degrees and will then call the appropriate math methods above, to return the sine, cosine and tan for an angle (note you will have to convert each degree to radians).

You can create a method, convertDegreeToRadians, which can be called to do the conversion, so you do have to replicate code.

The main program should call **mysin()**, **mycos()**, and **mytan(),in order to** print a trig table for angles from 0 degrees to maxDegrees (this number will be input by the user). You should prompt the user to input this maxDegree number in the range 30-45 inclusive. You should create an additional “input” method which will prompt this input this number from the user and return it to the main calling function. The main program will then loop to create the trig table. You should check for valid input from the user for the maxDegree (using if/else statement). Results should be formatted and printed in a tabular display with the values number aligned and values should be displayed up to 4 digits to the right of the decimal point (See example screenshot, showing trig table for 0-45 degrees.

Please submit one MS Word file containing a flow chart of your algorithm, along with your C# code (copy/paste) and testing screenshots.

A close-up of a document

Description automatically generated with low confidence