Name: _____

Sheet 451: Mini-Project: Program to Find a Zero on an Interval

Goal: Write an algorithm that finds the leftmost zero on a given interval.

Write a method for the function *f* that takes parameters *x*, (the variable) *a*, *b*, *c*, *d* (the constants) and computes any third degree polynomial:
$f(x) = ax^3 + bx^2 + cx + d$ and returns *f* as a double.
It should be changeable to other functions as desired.

A zero means that $f(x) = 0$.

Double variables (don't worry too much about constants vs variables at this point) are max, min, a, b, c, d.
Max and min are the bounds of the interval. You can assume (-1, 1), meaning $-1 < x < 1$, or min = -1.0 and max = 1.0. This means you search for a zero from the minimum bound to the maximum bound. You only need to report the leftmost zero in the bound. The precision is up to you, but you can begin with an answer to within 0.1. The variables can be initialized in the code but the code should be able to be modified so the variables a, b, c, and d are provided by the user (eventually also max, and min, but this is not required).

To find a zero, look for a zero crossing. That is, for when the function goes from negative to positive as x increases, or from positive to negative as x increases. When that happens, you have a zero. That's what the calculator does.

You can use your calculator to see what these values of a, b, c, d do. Use the values below to test your program. These assume min = -1.0 and max = 1.0 and a third degree polynomial. Test vectors just mean values passed to a program to see if it works in different scenarios.

```
/* Useful Test vectors:
1 0 -1 0; 1 1 0 1; 1 0 -0.5 0;
1 -0.8 1 -0.8; 1 -0.11 1 -0.11;
1 -1 1 -1; 1 -8 1 5;
-1 2 0 -0.9;  -1 4 0 -1;
0 1 -0.9 0.2025;
-1 -0.6 0.0675 0.06075;
1 -0.6 -0.0675 0.06075;
4 5 -1 -2; */
```

Tasks:

(1) Understand the concept. Use your graphing calculator to find Zeros with `[2nd] [TRACE/CALC] 2:zero`.
(2) Make a plan and communicate it to the teacher. Do not start coding without a general plan!
(3) Write some code and show it to the teacher.
(4) Try to make it work. (Keep a log of what does not work yet. It's OK if something does not work as long as you keep track of what does not work).
(5) Remove redundant code. Make sure you are using methods for repeated tasks.
(6) Figure out how to handle special cases.
(7) Increase the accuracy.
(8) Work on extensions: higher accuracy, different bounds, different functions, multiple zeros, or higher speed. Let the teacher know what you are working on if it's a different idea from these!

You have ___ classes to complete the project. You will need to spend about ___ hours outside class. If you have to spend more, then see the teacher. We don't want this assignment to take over your lives.

Grading rubric

		Points
Quality of code		5
Quality of algorithm		5
Design, Planning		5
Complexity		5
Difficulty		Extra credit
Originality		5
Correctness		5
Executability: does it run?		5
Code comments		5
Log of tests performed (i.e. different test vectors)		5
Positive attitude		5
Helpfulness to others		part of participation grade
Timeliness		5
Grade	out of 50	55 possible