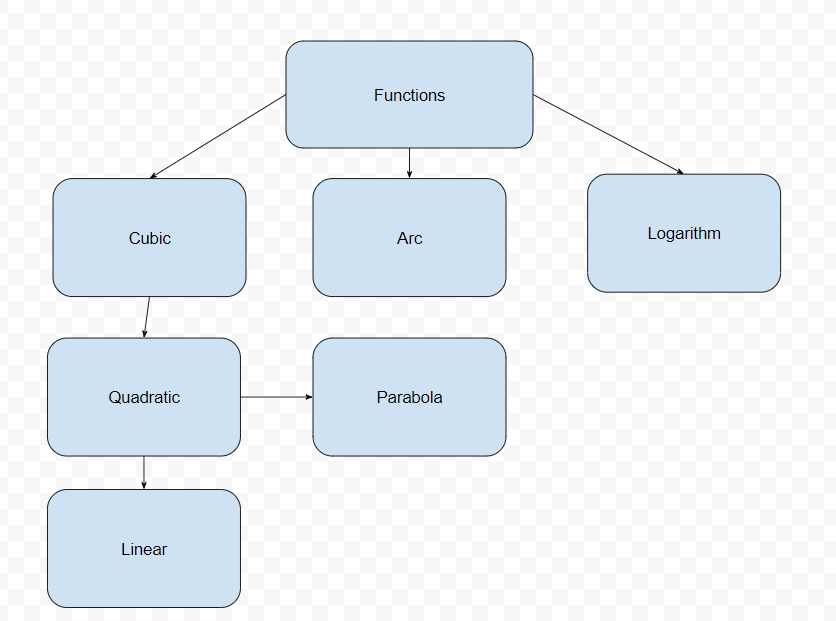
Limitations and Inheritance Structure

Assumptions: I assumed that the domain of the functions wouldn’t be very small (for example, 0.001) or very large (for example, 1,000,000).

Limitations: When drawing in the canvas, the labels that contain the coordinates may overlap with the function and/or the axis depending on the x and y increments. This happens because the x and y increments may contain a lot of decimal places.

Inheritance Structure:



A quadratic Is A cubic that has “a” = 0 in the equation y = a(x – x1)^3 + b(x – x1)^2 + c (x – x1) + d

A parabola Is A quadratic that has “b” = 0 in the equation y = a(x – x1)^2 + b(x – x1) + c

A linear Is A quadratic that has “a” = 0 in the equation y = a(x – x1)^2 + b(x – x1) + c

Therefore, the methods (calculations, drawable, and toString() ) from the Cubic class would also work for the subclasses (quadratic, parabola, linear) as long as you set the value of “a” or “b” to be equal to 0 inside the constructor. This means that only a constructor method will be required for the subclasses, since you can inherit all of the calculation, drawable, and toString() method from the Cubic class.

On the other hand, the Arc and Logarithm class are different functions with different types of restrictions. Adjustments would be needed for the methods. For example, toString() method would be different because the functions have sqrt() and ln() components respectively. Additionally, the undefined(double x) method would also vary, because you have to check if you are using sqrt() for a negative number and if you are using ln() for a non-positive number respectively.