Juan Villaman

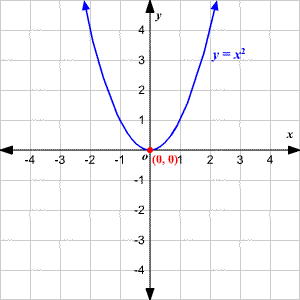
05/01/2025

**Plotter, Salter & Smoother in Java**

# What is a Plotter?

A **Plotter** is a program were, as its name states, plots data. For this plotter specifically, we are plotting this quadratic function:

This equation’s graph has the shape of a parabola (example below)



Source: [Varsity Tutors](https://www.varsitytutors.com/hotmath/hotmath_help/topics/parabolas)

I chose ArrayLists as my data structure to handle all of the data that’s being plotted. The program first shows the user the function, then asks the user to pick each coefficient a, b, and c. Using a for-loop, X-values and Y-values are added into their respective ArrayLists. I then took those values and stored them neatly into a CSV file using the PrintWriter class.

Below is a sample of the code asking for coefficient values and confirming that the data was written to our CSV file.

This is plotting:

A black background with white text

AI-generated content may be incorrect.

**This is the CSV file created by the code above**

A screen shot of numbers

AI-generated content may be incorrect.

**This is the function plotted in an XY graph**

## What is a Salter?

A **Salter** takes data and adds noise to it. What is noise? Noise is just a random number that is added or subtracted to our data. In this program, I salted the Y-values of our function. This is to simulate real world data, which may be random and be erratic when plotted.

My salting technique used is to set a range in which a number will be randomly selected for adding and subtracting to our Y-values. I used a Random object in java to randomly go between true and false, which determined if we add or subtracted from our Y-value. ArrayLists are still used to store our X and Y values of our function

This message shows the code confirming that our data has been salted, and those values have been stored in a CSV file.



**This is the CSV file created by the code above**

A screen shot of a black and white screen

AI-generated content may be incorrect.

**This is the function with its Y-values salted**

### What is a Smoother?

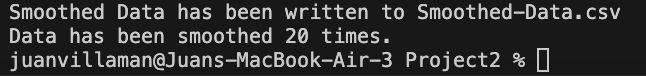
A **Smoother** smooth salted data for easier interpretation. In this case, we’re smoothing the Y-values we previously salted using our **Salter**. How exactly are we smoothing the data? We are using a **3-point rolling average**, also known as a **moving average**. This takes a point in our data, and calculates the average of its neighbor before it, and neighbor after it. For example, if we want to smooth a point , we will grab the neighbor before it (1, 8) and neighbor after (3, 17). Then we take the average of the 3 Y-values of those points. We smooth this point below:

By doing this to every point in our graph, we can effectively smooth the salted data. Our smoothed point in this case is (2, 11.6667). This process is done to every single point in the graph except the beginning and end points of our graph. With the beginning point, we only take the average of that point and its neighbor **after**. For the end point, we take the average of that point and its neighbor **before** it. Here is what it looks like implemented in our program:

A screen shot of a computer program

AI-generated content may be incorrect.

In this program, we can choose how many times we smooth the data to make it look as close as possible to the original values. For the function chosen, we smoothed the salted data 20 times.



**This is the CSV file created by the code above**

A screen shot of numbers

AI-generated content may be incorrect.

**This is the function with its Y-values smoothed**

As you can see, the points are now in smooth line instead the randomness in the salted graph. This helps us visualize data easier and be able to draw conclusions off the data we are studying.