

# Function Plot Project

## Description

Hello! We thank you for considering us for your internship, and we hope you enjoy this project! We designed this project to be a fun and simple way to showcase your software engineering skills. We will be evaluating your project for functionality, code cleanliness, and correctness. If you have any questions, please contact [marlon.calvo.ext@siemens.com](mailto:marlon.calvo.ext@siemens.com) or [sachin.shinde@siemens.com](mailto:sachin.shinde@siemens.com) for any queries.

At Siemens Energy, we heavily rely on a vast variety of numerical methods to perform our predictive analyses. Polynomial functions are one of the critical techniques we apply daily at Siemens Energy for performing regressions on data, fitting curves to models, and machine learning. In this project, you will utilize a multivariate polynomial function plotter to visualize some cool 3D charts!

Recall that a polynomial function,  $y = c_n x^n + c_{n-1} x^{n-1} + \dots + c_1 x^1 + b$ , is a function with  $n$  coefficients and  $n$  powers of  $x$ . Furthermore, a multivariate function, i.e. a 2D function  $f(x, y) = x + y$ , is one which takes two or more inputs. Furthermore, one can have a multivariate polynomial function, i.e.  $f(x, y) = x^2 + x + y^2 + y$ . For our specific program, let's assume  $f(x, y) = c_2 x^2 + c_2 x + c_1 y^2 + c_2 y + b$ . We will be generating different scatter plots with  $f$ .

## Project Details

We have built a program, called *engine*, which produces a collection of plot data points for each plot recognized in its input. It generates an output file "*output.out*" in the working directory with the following structure:

```
=== Plot for cool_plot ===
0,0,1.200000
0,1,101.200000
0,2,201.200000
...
<x>,<y>,<z>
```

Each row defines a cartesian point in 3D space computed from the coefficients  $c_2$ ,  $c_1$  and the bias  $b$ . This information can be used to plot this function in the project's 3D tool. This program recognizes CSV files with the following format to generate the plots:

```
cool_plot,name,3,-5,2
```

It can have any number of rows, but the columns are restricted as following: non-whitespace non-comma character name, number for  $c_1$ , number for  $c_2$ , number for  $b$ . Please avoid using any spaces for any columns in this CSV file. To run the program, simply call the executable, in your terminal, like so:

```
Windows: engine.exe plots.csv
Unix:    ./engine plots.csv
```

Both executables can be found in the "*src/main/resources/*" folder. Your task is to use the output from this tool to create the plots in your project.

The project requirements are as follows: complete the application to allow a command-line argument for the path to a *Function Plot* input file (described below) and generate the requested plots via the *engine* utility. You are expected to amend, or slightly modify, the existing codebase; it is suggested that you keep the project layout intact though. More specifically, there are sections of the code which must be implemented, which are: reading of *Function Plot* input, creation of *engine* input, running of *engine*, and reading output of *engine*. Remember to keep in mind our evaluation metrics as you design these methods.

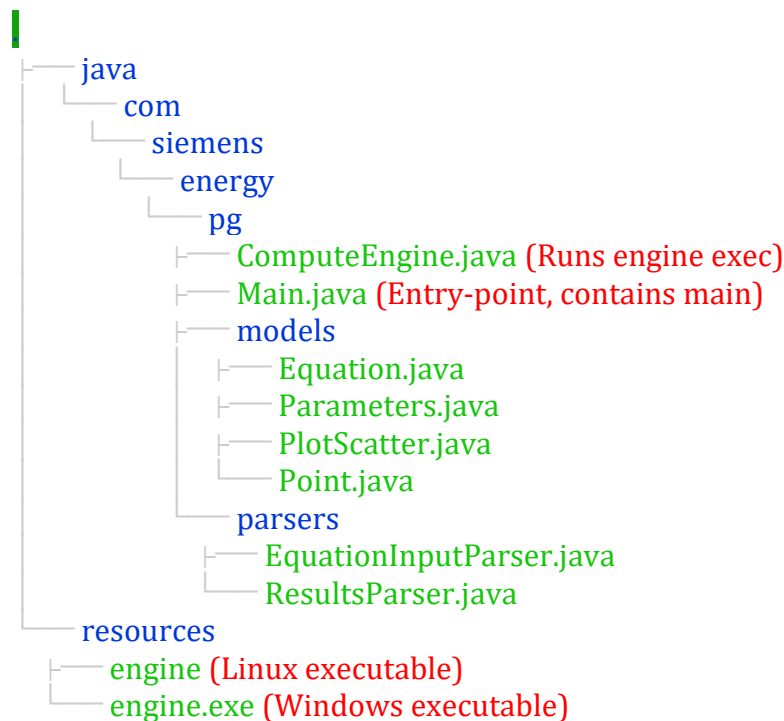
The input file structure for *Function Plot* is as follows:

Plot_1	Plot_2	Plot_3	...	Plot_N
C11	C21	C31	...	CN1
C21	C22	C32	...	CN2
B1	B2	B3	...	BN

To run the project, open a command-line window and type in the following:

Windows: `gradlew.bat run --args="example_files\input.in"`  
 Unix: `./gradlew run --args="example_files\input.in"`

The folder structure is as follows:



You can find examples of all the utilized and generated input/output files in the “*example\_files*” found in the project’s root directory.

Some useful information about this project, which is not required to understand:

- Gradle 6.3 is this project’s build system
- GUI framework is AWT

- Using Jyz3D API for 3D plotting

Useful resources:

- <https://docs.oracle.com/javase/8/docs/api/java/lang/Process.html>
- [https://en.wikipedia.org/wiki/Comma-separated\\_values](https://en.wikipedia.org/wiki/Comma-separated_values)