Project 1: Console-based Plotting

1 Objective

Implement a Java program that includes one class with multiple methods (each having parameters). Use standard output methods. Use definite loops. Create and use class-level constants.

2 Task

Write a program to do console-based plotting of curves/lines in quadrants I and II, but *sideways*. Write four functions (plotXSquared, plotNegXSquaredPlus20, plotAbsXPlus1, plotSinWave) plus a main program to drive them and feed them hardcoded data (since we have not yet covered user input). Each function should accept two integer parameters indicating the minimum and maximum X value to be plotted. The functions should have no return values. Create class-level constants for things we might wish to change later, e.g., a String called PLOT_CHAR that holds an asterisk (per samples below).

3 Output

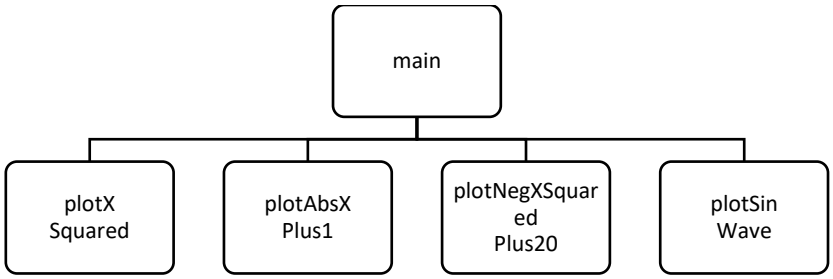
When functions are called with indicated X values, output should be as shown on the next page. In the functions, construct subtitles using passed-in data; do not hardcode ranges (we might change them later).

4 Code Implementation

Create a class called SidePlot with all functions within that class. Follow our Course Style Guide on Canvas.

4.1 Call Hierarchy

Create the following public static functions. Use this exact call hierarchy (i.e., second-tier functions must *not* call each other; that would be *chaining,* and we never want that):



4.2 What You **May** Use

- Constants (at class level), using the requested naming convention (see the Course Style Guide)
- Variables (but not class-level ones), using requested naming convention (camelCasing)
- Assignment and calculations
- Definite loops (for); note that each graph should require no more than *two* for statements
- Console output
- Additional helper functions to reduce redundancy

4.3 What You **May Not** Use

- Indefinite loops (while); they are not helpful or indicated here, anyway
- Selection control structures (if) unless you are doing the extra credit
- Arrays, lists, or other data structures, objects, libraries, or methods we have not covered

5 Output

```
Sideways Plot
y = x*x where -6<=x<=6
                                    *
                            *
                    *
            *
    *
 *
*
 *
    *
            *
                    *
                            *
                                    *

Sideways Plot
y = |x|+1, where -5<=x<=5
      *
     *
    *
   *
  *
 *
  *
   *
    *
     *
      *

Sideways Plot
y = -(x*x)+20 where -4<=x<=4
    *

           *

                *

                   *

                    *

                   *

                *

           *

    *

Sideways Plot
y = 20sin(.5x)+20 where -9<=x<=9
                                        *
                                    *
                            *
                    *
        *
 *
*
   *
          *
                  *
                          *
                                  *
                                      *
                                    *
                              *
                        *
             *
    *
*
```

6 Extra Credit A

Instead of the output shown above, produce this output which includes axis markers and a glimpse into quadrants III and IV. You are allowed selection control structures for this work. Note that the asterisk sits slightly above the hyphen on the output; choose another plot character (e.g., "o") if this bothers you.

```
Sideways Plot
y = x*x where -6<=x<=6
  |                                    *
  |                           *
  |                   *
  |           *
  |   *
  |*
--*-----------------------------------
  |*
  |   *
  |         *
  |                 *
  |                           *
  |                                    *

Sideways Plot
y = |x|+1 where -5<=x<=5
  |     *
  |    *
  |   *
  |  *
  | *
--+*------
  | *
  |  *
  |   *
  |    *
  |     *

Sideways Plot
y = -(x*x)+20 where -4<=x<=4
  |    *
  |          *
  |               *
  |                  *
--+-------------------*-
  |                  *
  |               *
  |          *
  |   *

Sideways Plot
y = 20sin(.5x)+20 where -9<=x<=9
  |                                        *
  |                                    *
  |                               *
  |                  *
  |       *
  |*
  *
  |  *
  |         *
--+-------------------*--------------------
  |                              *
  |                                    *
  |                                       *
  |                                     *
  |                               *
  |                         *
  |           *
  |   *
  *
```

7 Extra Credit B

Write a separate class called Limits with a main function that uses definite loops to verify this statement:

> *For a loan of $10,000 at an annual interest rate of 3%, the limit of the total amount paid as the number of payments approaches infinity is $10,150.75*

Output should start like this:

```
# Pmts Total Paid
1      10300.0
10     10165.74
100    ...
```

To accomplish this task, you will need several building blocks:

- The formula for calculating a loan payment amount is shown below. We would normally round to the nearest penny—you cannot make payments in fractional cents—but leave it at full precision instead

$$P = \frac{r(PV)}{1-(1+r)^{-n}}$$

$P = Payment$
$PV = Present\ Value$
$r = rate\ per\ period$
$n = number\ of\ periods$

- The formula for calculating the total amount paid: TP = P(n)
  - Round this to the nearest penny before display
- A Java method that rounds to the nearest integer: Math.round(*number*)
  - Note that there is no second parameter; you will need to get creative
- A Java method that raises a number to a power: Math.pow(*number*, *exponent*)
- Incrementing your loop variable logarithmically; we want to see the number of payments and the total of payments for 1 payment a year, 10 payments a year, etc., through 1,000,000,000 payments a year (10^9).

8 Submitting Your Work

Submit the source code (.java file(s)) from your project; there is no need to send other files or zip up the folder. Submit all files in a single Canvas submission (using the provided option to add another file).

9 Hints

There is a good bit of duplicated code between the plotting functions; do not worry about it in this assignment. If you are clever, you might create a helper function or two that might reduce duplication, e.g., to help build the display header for each graph.

You do not need to guard against bad data coming in via parameters (in later work, you will). But you should test your program with values that are in the specified ranges that should work.

10 Grading Matrix

Area	Points
Class	5%
Functions	75%
Main	10%
Documentation/style	10%
Extra credit A	5%
Extra credit B	2%
Total	**107%**

11 Suggested Plan of Attack

This should be a straightforward project to code.

1. Carefully read the entire project. Pay careful attention to sections on what you may and may not use.
2. Create the requested global constant (the string to use for plotting).
3. Code each plotting function, one by one.
   a. Consider copying method names from the project PDF to ensure correct spelling and casing.
   b. Add a call to the completed function in main() to check output.
   c. Remember that arguments may contain different values; your code must be flexible.
4. Check output with different arguments in the method call in main(); ensure proper output in each case.
5. Do a style pass, referring to the course Style Guide. Consider asking your IDE to reformat your code; this fixes most indentation issues.
6. Double check your output against the sample output; it should look the same.
7. Again, read the project in detail; use it as a checklist to ensure completeness and correctness of your code.