Notes:

First, maybe check if the syntax for the program is correct. So first check for empty input. Next check if all the parenthesis are closed at some point (in the correct order). Maybe then go to the very inner parenthesis; so check if you have parenthesis. Next check if there are parenthesis within the parenthesis (make sure they are not empty also). Then repeat; so if there are parenthesis within those parenthesis (again non-empty). This continues until there is not another level down (note: you would backtrack and then try the other possibilities later). When you find that current furthest point down, you need to compute (method compute) whatever is in the parenthesis. Note, it would be good to check the syntax each time whenever you call the compute method.

Note: to check syntax, need to check binary operators (+, -, /, \*, ^). This means that whenever you see on of those symbols, you need to check both sides of that symbol. On one side (left) there needs to be either a closing parenthesis “)” or a number. (note: maybe don’t need to check for parenthesis? Because will be at the furthest point down—seems like is true). Same would go for the other side. There should be a number on the right side of the operation. So I need to probably parse things by the symbols first. For example: 20+30-10

Here, I need to parse like so: 20 30 10

Note: can make a tree? To save the sequence of operations

So basically after find what operations to compute, then you need to execute them and return a number. Once that is done, then it needs to be substituted back into the formula. This would then continue until there is one final result.

Note: unary operators have highest precedence

Note:

(((3))) this is mathematically correct.

Version control—learn to use it; paragraph why you are using this one; advantages over the other ones. (this is the one you are using for your main project—I think; would be good to include if so). Also, need screenshot of all the pushes that you added to github

Ghent chart: include milestones, dates, (10-15min)

Note: need to check special cases such as divide by zero