Changes:

I removed the parse class, since it was small enough to fit in a function. This removed and extra file.

I created a inner the stack as an inner class, rather than a seperate class. This was similar to the parse class in which this removed an unecessary class file.

I created a Handler class called BigIntNumber. This class removed the need for the calculator to keep track of whether the number was the left most item (Called Top in the code), and whether it is negative. It also retrieved the size of the number for comparing the two numbers.

File reader was given more funcionality. It retrieves the next char, and records the value into a string. Since the parser ingores unnecessary symbols, the file reader can keep track of the original equation better. The file reader needs to return the created string as well, and erase it for the next equation that is loaded.

The operations class was created to remove some bulk from the calculator class. It handles what operations need to performed, and what special conditions are present. Special conditions include both positive, one negative, and both negative. In addition each needs to handle whether the first number is bigger or the second to perform certain operations properly. With out these hanlder, we could get crazy numbers like 1 - 10 = -11 instead of -9.

Calculator was made more simple. It launches the methods to make the program run, rather then performing all the functionality. It hold need information like the numbers, file reader, and whether me need to subtract or not.

What: I learn:

You know more about the program when implementing (what works and what doesn't), so you should aways be opened to change.

It is important to visualize your problem. You should identify special cases that may occur in your program, also identify limitation in the the tools you use.

Planning is very important. It starts you off in the right direction. Even if you original design does not work, a Simple refactoring can solve your issues. If you go in blid, you can risk having to rework your code from the ground up.