**A brief description**

**Implementation of CCAL.g4**

I made a grammar file called “CCAL.g4” here I made my rules and assigned variables. I created parser rules, added in the alphabet using upper and lowercase, keywords, operators, tokens, complex tokens, whitespace, and comments. These were the building blocks in the code.

**Implementation of CCAL.java**

After making the grammar file, “CCAL.g4”. I ran the antlr4 command using my grammar file to see the results of the testing for my grammar using the grun command to ensure my rules were correctly. I would see the syntax tree appear. I began working on my CCAL.java file which is the semantic analyser, I used this to create a parser that checked whether a “.ccl” file was successfully parsed or not. After thorough research I was able to finish the semantic analyser. To run you would need to compile first ‘compile CCAL\*.java’ and then enter in ‘java CCAL file.ccl’ for the analyser to work.

[**https://www.willowtreeapps.com/craft/an-introduction-to-language-lexing-and-parsing-with-antlr**](https://www.willowtreeapps.com/craft/an-introduction-to-language-lexing-and-parsing-with-antlr)

[**https://tomassetti.me/best-practices-for-antlr-parsers/**](https://tomassetti.me/best-practices-for-antlr-parsers/)

[**https://www.aldoraweb.com/antlr-error-handle/**](https://www.aldoraweb.com/antlr-error-handle/)

[**https://snyk.io/advisor/npm-package/antlr4/functions/antlr4.error**](https://snyk.io/advisor/npm-package/antlr4/functions/antlr4.error)