COP4020 Homework Assignment 4

State the question prior to answering.

1. Write a regular expression to capture the format of floating point constants in C/C++.
2. Given the unambiguous LL(1) grammar for simple expressions E shown on the lecture notes on Syntax, construct the parse tree of `2*a/(b+1)`
3. Textbook exercise 2.4 (a)
4. Textbook exercise 2.13 (b), (c), and (e)
5. Textbook exercise 2.18. Hint: to show that the grammar is not LL(1), try to construct a recursive descent parser. Note that *term* can start with an `id` (via *factor*) and so does the right-hand side of the first production of *expr*. A recursive descent parser for an LL(1) grammar has a lookahead of only one token. So what happens when you parse `a+b` and `a:=0` ? How can you fix this?