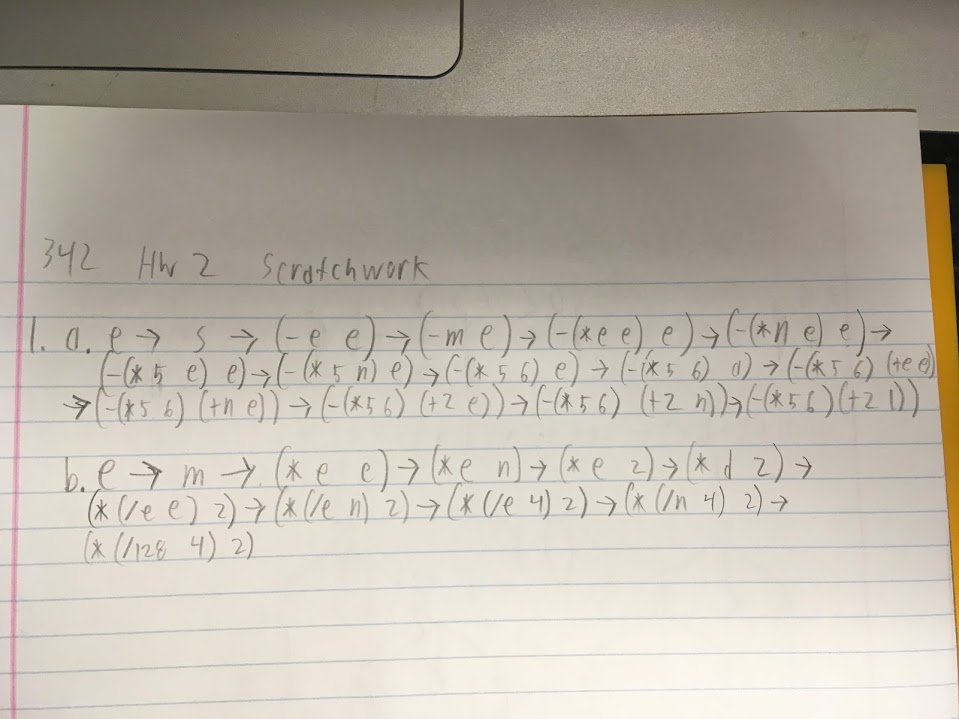
Homework 2

Christian Shinkle

1. j
   1. j
   2. j



1. j
   1. j

* AST.java defines the interface for the Visitor pattern the interpreter will be using. It also contains the code for all the subclasses of Exp.
* Evaluator.java contains an implementation for all the visit methods for the Visitor pattern. This means it contains all the instructions for computation of the expressions.
* Interpreter.java contains the main method for interpreter. It is the implementation of the Read-Eval-Print-Loop using the Reader, Evaluator, and Printer classes.
* Printer.java prints out the corresponding symbols and expressions according to the type of expression passed in.
* Reader.java parses the input of the interpreter, determines if the input is valid, and prints the output if it is.
* Value.java is just a wrapper class to help print out numerical values using its toString method.
* ArithLang.g contains the grammar for ArithLang.
  1. The production rule for addexp starts on the first line by saying it will return an AddExp. The next line says there is a local variable for an ArrayList called list that is passed in. The next line says list will be initialized before the start of the production rule. After the colon, the expression returned by the rule will start with the two characters ‘(‘ and ‘+’. Then the next line says an expression e will be concatenated, followed by another expression. Both of these expressions will be added to list. Then ‘)’ will be concatenated to the end of the expression. Lastly, list will be passed on to the next expression just as the previous list was passed in.