# Hunter Berry

ULID: hpberr1

# 

# 

# 

# 

# Programming Assignment 2

# LR1 Parser in Java

Secret Dir: IT327PROG2!

## Program Summary

Data structures:

* ArrayList
  + This acts as the queue for the program
* Stack
  + Holds stackObj objects
* StackObj - class for custom object to hold different data
  + Fields
    - Nonterminal symbol - optional
    - Terminal symbol
    - State
  + ToString method that prints in the form (T=5:9) or (\*:7)

Methods:

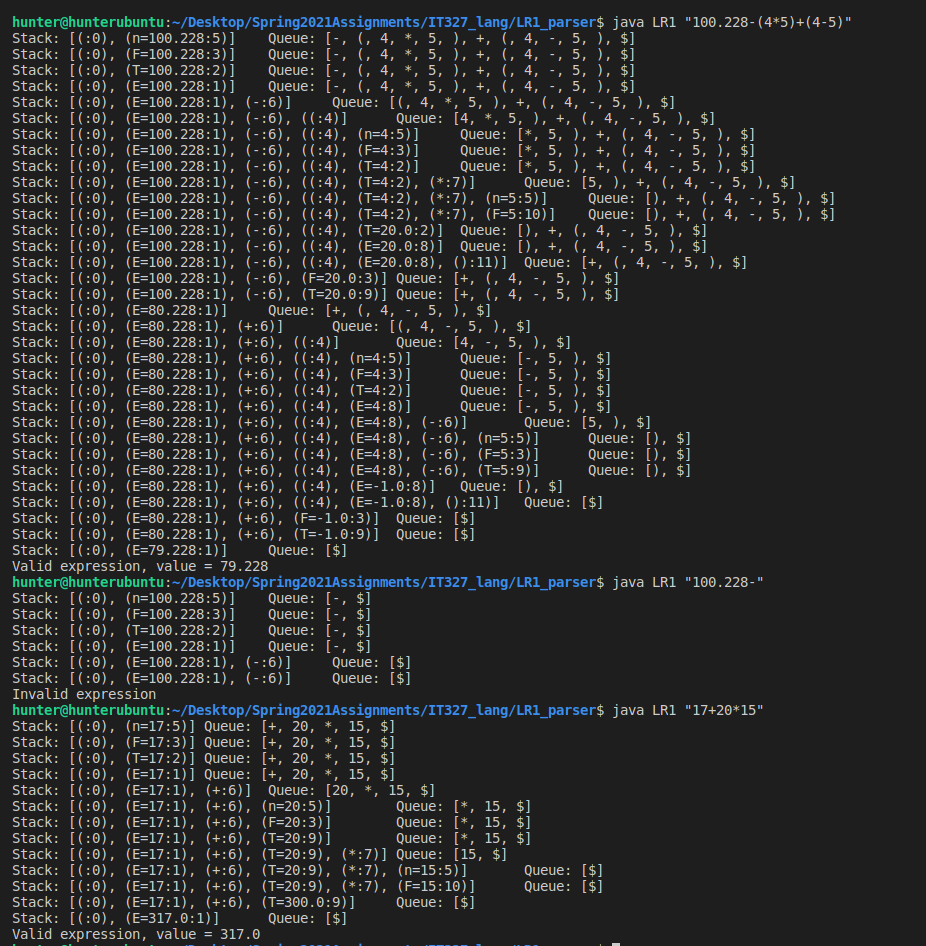
* Boolean parseTable()
  + a 2d switch statement to represent the table and the rules
* Int nonTermTable
  + 2d switch statement to represent the nonterminal table and the states
* Void shift(int state)
  + Dequeue input element and push to stack
* Void reduce(String newNT)
  + Pop from stack and change nonTerminal and state then push back to stack
* Void reduceOperation()
  + Pop three elements from the stack and perform operations on them.
  + Push result of operation to stack
* StackObj calc(StackObj first, StackObj op, StackObj second)
  + Calculate the result object with new state/nonterminal

Main method:

* Tokenize input
* Initialize stack and queue
* Enter main loop that continues until parseTable returns accept state

For this program I read the input using string tokenizer and I used an ArrayList to hold the tokens in a queue. Enters a while loop that continuously calls parseTable() on the inputs until the parseTable method returns true. I didn’t encounter any problems while making this program. I wrote out all of the methods I would need and designed the stackObj class before starting to code to avoid problems later on and it seems to have worked since I didn’t encounter any bugs. I also wrote a bash script that tests the output of the program on the evaluation of some arithmetic expressions.

Example input and output of program:



LR1 testScript:

