# Assignment 4
**Due: July 19**

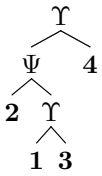
## 1 Goals

1. Practice building parse trees and ASTs
2. Practice evaluating ASTs
3. Practice defining a grammar based on given rules

## 2 Exercises

Write out solutions to the exercises neatly on another sheet of paper, following the grammar and tree formatting in the lecture slides. Make sure to include your name on the assignment. Submit this **in person at the beginning of class** on the due date.

1. Convert the following AST to RPN:




2. Given $a \Upsilon b \equiv 2a + b$ and $a \Psi b \equiv a^2 - b$, evaluate the numeric result of question 1.

3. Given the grammar with `<exp>` as the start symbol:

```
<exp> ::= <exp> + <mulexp> | <mulexp>
<mulexp> ::= <mulexp> * <rootexp> | <rootexp>
<rootexp> ::= (<exp>) | a | b | c | d | e
```

   Attempt to draw parse trees for the following sentences, working from the bottom up. Apply as many productions as you can, and denote if a sentence is invalid.

   - `((a*e))+b+(b)`
   - `c + (a)(d+e)`
   - `a*b*c+(d+e)*a`

4. Write unambiguous BNF grammar rules for a subset of Java-style method calls. The grammar should obey the following rules:

   - A method can take any number of parameters
   - All parameters are comma separated
   - Each parameter can be a variable, a method call, or a simple arithmetic expression
   - A simple arithmetic expression can be variables/method calls separated by + and * only – no parentheses or other operations
   - Comma-separated and arithmetic expressions are evaluated left-to-right
   - A method call has higher precedence than arithmetic
   - * has higher precedence than +

- Variables can only be named `x`, `y`, `z`
- Methods can only be named `m1`, `m2`, `m3`
- Nested method calls must be evaluated from the inside out – in `m1(m2())`, `m2` should be evaluated before `m1`
- The start symbol is a method call

Write the rules using the formatting from the slides, and make sure you specify the start symbol.

Some legal examples in the language:

```
m1(x, y, z)
m2(m1() + m3(), x * m3())
m1(m1(m1(m2(x+y))))
```

Some **illegal** examples:

```
m1(x,)
1 + m2()
m1(m2 * y)
```

5. Draw a parse tree **and** AST for each of the following expressions, using your grammar. For AST syntax, treat a method node like an operator node, where its parameters are analagous to operands:
   - `m1(x*y)`
   - `m3(x+m2(z,z), y)`
   - `m2(m2(m2(x, m1())))`

   Example AST for `m1(x,y,z)`:


## 3 Grading

Exercises have the following weights:

- Exercise 1: 10%
- Exercise 2: 10%
- Exercise 3: 20%
- Exercise 4: 40%
- Exercise 5: 20%