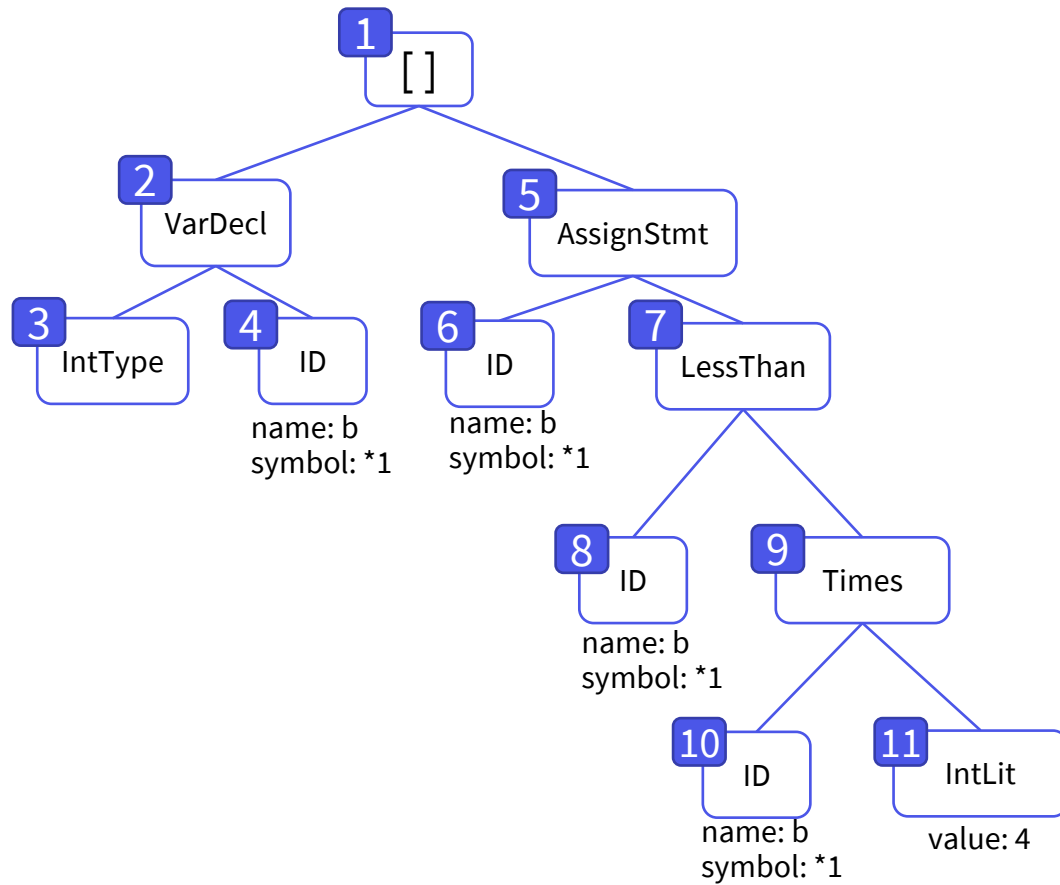




# Check-in

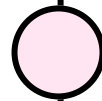
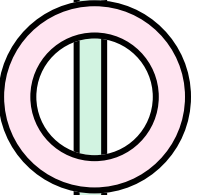
Assume a program snippet has generated the following AST. Annotate each node with the type to which it corresponds (or error if it is an error type). If a type analysis would issue a report, indicate that as well



- 1 \_\_\_\_\_
- 2 \_\_\_\_\_
- 3 \_\_\_\_\_
- 4 \_\_\_\_\_
- 5 \_\_\_\_\_
- 6 \_\_\_\_\_
- 7 \_\_\_\_\_
- 8 \_\_\_\_\_
- 9 \_\_\_\_\_
- 10 \_\_\_\_\_
- 11 \_\_\_\_\_



**FLIPPED  
WEDNESDAY**



# ○ Administrivia

- You should have gotten the midpoint survey, I'd really appreciate if you fill it out!



# ○ Written Work #6

## **Topics:**

- SLR Parsers
- Scope
- Semantic Analysis



# ○ Written Work #6: Question 1

Recall that Jeff is a statically-scoped language. Assume a new variant of Jeff, called Jeffrey, that uses the exact same syntax as Jeff but is dynamically-scoped. Provide an input file that would be valid in both Jeff or Jeffrey', but prints a different result based on which scoping system is used. Show what the output would be either under scheme.



# ○ Written Work #6: Question 2

Provide an input file that would be valid in Drewgon', but **not** in Drewgon.



# ○ Written Work #6: Question 3

Explain the difference between an SLR and an LR(0) parser. Provide an example grammar that is SLR-parseable but not LR(0) parseable.



# Written Work #6: Question 4

Draw the symbol table at the point in name analysis after line 11 but before line 12.

```
1. void func(int arg){
2.     int a;
3. }
4. void funky(int arg){
5.     int l;
6.     if (l > 2){
7.         int m;
8.         m = l;
9.     }
10.    if (l < 2){
11.        int n;
12.        n = l;
13.    }
14. }
```

