CSCI_4230_PL_10_22_2018.md

Midterm talk. Scheme and EBNF/BNF stuff abound!

```
data BinTree a = Empty | Node a (BinTree a) (Bin Tree b) :-- left and right
tree...

myTree = Empty

:t myTree

:-- prints "myTree :: BinTree a"

myTree = Node "root" (Node "left" Empty Empty) (Node "right" Empty Empty)

:t myTree

:-- prints "myTree :: BinTree [Char]"

data Element = Element {element_name :: String, atomic_number :: Int, atomic
weight :: float}

mercury = Element {atomic_number = 80, element_name = "Hg", atomic_weight =
208.592}

atomic_weight mercury

:-- prints "200.592"

element_name mercury

:-- prints "\"Hg\""

atomic_number mercury

:-- prints "80"

:t mercury

:-- prints "mercury :: Element"
```

Pattern Matching

```
atomic_number (name, number, weight) = number
mercury = ("Hg", 80, 200.592)
atomic_number mercury

:-- prints "80"
```

```
atomic_number (5, True, "Hello")

:-- prints "Two"

:t atomic_number

:-- prints "atomic_number :: (a,b,c) -> b"

second_of_three = one, two, three = two

:t second_of_three

:-- prints "second_of_three :: p1 -> p2 -> p3 -> p2"
```