1. Slideshow
    a. Higher-order functions in Scala
        i. What are higher-order functions?
            1. Importance in programming
            2. Real-world use cases
            3. How to write a higher-order function
        ii. foldLeft/foldRight
            1. Using foldLeft/foldRight to recurse over trees
                a. Using tree data structure
                b. Using tree data structure with nodes being list of integers (links to map)
                c. Time complexity/real world use cases
        iii. Map
            1. Summing list of integers within tree data structure (links to foldLeft/foldRight)
            2. Time complexity/real world use cases
            3. Look into parallelism
        iv. flatMap
            1. Different uses over list
            2. Different uses over tree
            3. Special case with tree containing lists
            4. Time complexity/real world use cases
        v. mapFirst
            1. Using tree data structure
            2. Using tree data structure with nodes being list of integers (links to foldLeft/foldRight)
            3. Time complexity/real world use cases
    b. Best languages for higher-order functions
        i. Rust vs. Scala?
        ii. Can this really only be implemented in functional programming?
    c. Possibly have interactive last part where students can create their own higher-order function

Notes
- Make our own list class