CSE 143 Review:

* L1 & L2 - Irrelevant
* L3
  + Traversal
    - For loop
    - For each loop – read only
    - Iterator (Iterator<E> itr = this.iterator()); - no concurrent modification – remove q on final
      * hasNext() – checks if has next
      * next() – gets next
      * remove() – removes most recent val
* L4
  + Stacks & Queues
    - Stacks: LIFO
      * Push/pop
      * Stack<E> s = new Stack<>();
    - Queues: FIFO
      * Add/remove
      * Queue<E> q = new LinkedList<>():
* L5
  + Interfaces
    - Contains behaviors(methods) w/o body
    - Use *implements* keyword
    - You can implement multiple interfaces (cannot inherit multiple classes)
* L6, L7, L8
  + LinkedLists
    - Manipulating list using references
    - Iterating thru list – for various reasons – check all cases: front, end, middle, etc.
    - Overloading
    - This notation
* L9
  + Binary Search Complexity
* L10, L11
  + Maps ~ TreeMap
    - Methods:
      * Put(key, value)
        + 1 key : 1 value
      * Can print map
      * Adding depends on whether value exists or not (for list structures)
      * containsKey(key)
      * Keyset()
      * get(key)
      * remove(key)
* L12
  + Recursion
    - Base case
    - Recursive case
* L13
  + Regular Expressions
* L14
  + Inheritance