Final Exam Objectives !!

In order to successfully complete this final, you must be able to:

(Java objects and heirarchy)

- Explain the difference between IsA and HasA relationships
- Write subclasses for abstract classes
- Explain the difference between an overwritten method and an abstract method
- Explain the difference between an interface and an abstract class.
- Explain the relation between the Object class and other classes.
- Explain when a `.equals()` method must be used instead of a == operator.
- Write a `.equals()` method.

(Queues and Stacks)

- Define and explain the Queue and Stack Abstract Data Types
- Identify applications of Queue and Stack Abstract Data Types
- Write array based and linked-list based implementations for methods that check if a Queue is empty, that enqueue to and dequeue items from a Queue
- Write array based and linked-list based implementations for methods that check if a Stack is empty, that push to and pop items from a Stack
- Indicate and defend the running time of each of above Stack and Queue methods

(Exceptions)

- Write exception classes
- Call methods of exception classes
- Throw and Catch exceptions

(Binary Trees & Heaps)

- determine the height of a specified binary tree
- explain the difference between full, complete and balanced binary trees
- write array based and reference based implementations of binary trees, including methods for: creating, inserting, and finding.
- for each implementation above, be able to traverse binary trees in pre-order, in-order, post-order.
- Indicate and defend the running time of each of above traversal methods.
- write array based implementations of a heap, including methods for: creating, inserting& deleting.
- Analyze array and reference based based implementations of a heap, including methods for: creating, inserting& deleting. In particular, be able to explain why the big O running times of the heap insert and delete methods are O(log N). (Hint: the argument should be based on the relationship between the number of nodes in a full tree and its height).
- Indicate and defend the running time of each of above Tree methods.

(Linked Lists and Resizable Arrays)

- Trace and write linked lists, including singly and doubly linked lists and circular lists
- Be able to create linked lists and insert into, delete from and traverse them. Determine the big O of each operation.
- Be able to create resizable arrays and insert into, delete from and display them. Determine the big O of each operation.

(Efficiency)

- Compare and contrast different algorithms that accomplish the same goal. This includes algorithm efficiency.
- Given a piece of code or pseudo code, analyze using the big O notation
- Compare the growth rates of various algorithms

(Recursion)

- Trace and write recursive functions.
- Compare efficiency of recursive versus iterative algorithms

Also: review to the midterm exam objectives: Those could be tested too!

Format

- No electronic devices (including phones or calculators) will be permitted.
- Expect the exam will consist of questions that require you to:
  - Write small code sequences
  - Analyze code to determine its effects
  - Analyze code to determine its correctness
  - Separate correct and incorrect statements