

ETEC2101

Final Exam Important Study Concepts

Final Exam is 12/12/2016 (Monday) from 12:00 – 1:50pm

Use this study guide at your own risk. I'm just trying to list the most important concepts from the semester here. But everything from the entire semester is fair game.

Don't forget to get a good night sleep and to each lunch before the test – it really helps 😊

1. Intro to Java
 - a. basic java (variable declarations, System.out.println, class structure)
 - b. Use of iterators
 - c. Scanners (or equivalent forms of reading user keyboard input).
 - d. "normal" arrays in Java
 - e. static vs. instance variables
 - f. access modifiers
 - g. packages
 - h. bubble-sort
2. LinkedLists
 - a. singly vs. doubly linked lists.
 - b. nested classes
 - c. implements and extends inheritance
 - d. major linked list methods and how they are implemented and Big-O of each:
 - i. addToBeg/End
 - ii. insert
 - iii. findIndex
 - e. LinkedList implementation of Stacks and Queues
3. Slick2D intro
 - a. has-a vs. is-a implementations
 - b. Stacks and Queues
 - c. Slick2D (I'm more interested in the inheritance and general structure than specific commands)
4. Binary Search Trees (BST)
 - a. Major terminology
 - b. Major methods and their implementation of BST
 - i. add
 - ii. find
 - iii. rebalance
 - c. Recursion in general
 - d. Recursion in BST's
 - e. Visiting nodes (pre-order, post-order, in-order)
 - f. enums in Java
 - g. generics in Java (and limiting to specific types of generics (like Comparable))
5. HashMaps and HashSets
 - a. What is a hash function (and what makes it good)
 - b. Collisions
 - c. Load Factor
 - d. HashMaps

- i. The idea of Key => Value pairing
 - ii. Linear probing and bucket approach for handling collisions
 - iii. Major method implementation and Big-O:
 - 1. set / add
 - 2. get
 - 3. remove
 - e. HashSets
 - i. How we leveraged our HashMap to make a HashSet class
 - ii. union, intersection, different implementation and uses.
- 6. Graphs
 - a. abstract classes and methods
 - b. major terminology
 - c. how we used generics
 - d. The “compact” representations of adjacency lists and adjacency matrix.
 - e. Breadth-first and Depth-first traversal of.
- 7. Heaps
 - a. purpose of the data-structure / algorithm (PriorityQueue)
 - b. big-O and implementation of major methods
 - i. push / add
 - ii. pop
 - iii. find / contains
 - c. How is it different than BST's?
 - d. connection between logical and internal representation
 - e. binary <-> decimal conversion (and connection to Java integer types)
 - f. bit-shifting.
- 8. Quadrees
 - a. Brute-force approach to the problem we were trying to solve
 - b. Relationship between buckets, children, and area (rectangle)
 - c. add algorithm
 - d. findOverlaps & collisions algorithm