

## CIS 351 – Final Exam Review

The best thing to do to prep for the final is to review and understand all the labs, quiz and Homework, and in-class activities that we have done throughout the semester. Pay particular attention to different terminologies. Make sure you can apply the term to examples of features in code. You should be able to trace code including code that involves more than one class. You should be able to identify different kinds of statements or other items in a provided class. You should be able to correctly evaluate an expression.

There will be at least few questions, where you will have to write CODE to solve a problem. You will probably be given some code and you will need to fill in other code based on a description of the method, section, or class. If you feel that your coding skills are weak or that it takes you a while to figure out how to approach the problem, use the zybook challenge and practice problems to get more practice. Make sure you understand how linked list, stack, queue can be used to solve different problems, for example using stack to evaluate arithmetic expression. Or using linked list to build a stack and perform push and pop operations on them. Pay very good attention to all the concepts that we have learned for trees, heap, binary search tree, Hashtable. Try to work with example dataset to sort it with different kind of sorting techniques. Finally, think about how the worst-case timing, BigO, can be determined for various operations.

### Topics for the final exam

1. Everything that is covered till Midterm Exam 1
2. Everything that is covered till Midterm Exam 2
3. Recursion and recursive function
  - a. How base case is determined
  - b. How recursive function are constructed
4. Trees:
  - a. Tree depth, height, size
  - b. Properties: full, complete, perfect
  - c. Balanced binary trees
  - d. Recursive Tree traversing technique
    - i. Preorder, inorder, postorder - how to do them recursively
  - e. Non-recursive tree traversing
    - i. Preorder, inorder, postorder - how to do them non-recursively, using stack and queue
  - f. Building tree from arithmetic expression
  - g. How to evaluate arithmetic expression using tree traversal
  - h. Breadth first search and Depth firsts search
  - i. Binary search tree and its properties
  - j. How to insert, delete and search in a binary search tree
5. Linear search and Binary search algorithms

6. Heap
  - a. Heap order property
  - b. Heap structure property
  - c. How to insert, delete and search in a heap
    - i. Percolating up and percolating down
  - d. How heap can be constructed
7. Hashing:
  - a. How hashing works
  - b. Hash functions
  - c. Collision, rehash, clustering, load factor
  - d. Collision resolution
  - e. Hashing with buckets
  - f. Hash table resizing
8. Sorting:
  - a. Bubble sort
  - b. Insertion sort
  - c. Selection sort
  - d. Merge sort
  - e. Quicksort
  - f. Heap sort
9. Java API's:
  - a. Collection
  - b. Set
  - c. TreeSet
  - d. Map, HashMap, TreeMap
  - e. Comparator