**Test 2 review**

- Topics covered:
  - Inheritance
    - What does a derived class inherit from the base class, what does the derived class not inherit from the base class.
    - The 3 data access types in a class: public, private, protected
    - The 3 kinds of inheritance: public inheritance, protected inheritance and private inheritance
    - Override member function
    - What happens when a derived class object is created and destroyed
    - Be able to write code to implement member functions in a derived class
    - Be able to trace program involving base and derived classes
    - Early vs late binding (dynamic binding)
      - When is early binding performed?
      - When is late binding performed?
      - Virtual function
      - What is Polymorphism in C++? How is it achieved?
      - Be able to trace program involving polymorphism
      - Understand the rules for declaring virtual functions

  - Algorithm Analysis
    - Time efficiency and memory efficiency
    - How do we determine the size of a problem N?
    - Best time, worst time, and average time complexity of an algorithm
    - Be able to analyze a code segment and derive the growth rate function f(N)
    - Be able to derive the Big O function given a growth rate function
    - Understand the order of growth of the functions

      $O(1) < O(\log N) < O(N) < O(N\log N) < O(N^2) < O(N^{2\log N}) < O(N^3) < \ldots < O(2^N) < O(3^N) < \ldots < O(e^N)$

  - Recursion
    - Be able to trace a recursive function
    - Be able to write recursive function given recurrence relation

  - Sorting Algorithms
    - Understand each of the sorting algorithms discussed
      - Know step by step how each sorting algorithm put data into sorted order
    - Understand how to analyze each sorting algorithm in terms of growth rate function and big O function

    - Heap
      - Heap Up and Heap Down operations
      - Understand the time complexity of the Heap Up and Down operations
      - Build a heap from an array of values
      - Perform Heap sort on an array of values
      - Understand the time complexity of Heap Sort

- Binary Tree
  - Tree height, level of nodes, parent node, ancestor nodes, leaf nodes, etc.
  - Full binary tree, complete binary tree, perfect binary tree
  - Balanced binary tree
  - Binary tree traversal : inorder, postorder, preorder.

- Binary Search Tree
  - Understand how each operation is performed and be able to write code for each operation:
    - Search for records or retrieve data from BST based on key value
    - Insertion new records into BST
    - Deletion records from a BST
    - Deallocate all nodes in a tree
    - Create a copy of an existing tree
  - Save the BST and rebuild the tree with minimum height
  - Compute the height of a BST
  - Assign levels for each node in a BST

- Study the homework problems
- Study the lecture notes
- Come to office hours to ask questions about any concepts or topics you are not clear about