Topics List for Exam 1

1. Material from CISC 3115

    - Classes

        - accessors and mutators (getter / setters)
        - private / public
        - Constructors and overloaded constructors
        - Overloading / overriding
        - How to override the equals() method
        - How to override the toString() method
        - How to implement the Comparable interface and compareTo()
        - interfaces and how to create / implement them
        - inheritances and polymorphism
        - extends and implements
        - Difference between declaring a reference and creating an object (operator new)

    - Recursion

        - How to write a simple recursive algorithm
        - How to trace through a recursive algorithm

2. Java Collections Hierarchy

    - What is it?
    - to know that it contains both classes and interfaces
    - to know that the broadest interface is called Collection
    - to know what the List interface is and its methods
    - to know what an ADT and an API are.

3. ArrayLists

    - How to create an empty ArrayList in Java
    - What the operations add(), size(), get(), set(), iterator(), listIterator(), remove(), contains(), etc. (all of the various List operations) do and to know how to use them to write a method/program segment.
    - To write a method that will iterate through an ArrayList performing some operation. You must know all 3 ways we looked at!
    - Our Own Implementation
        - How did we implement an ArrayList?
        - What is the running time of all of the ArrayList operations based on our implementation.

4. LinkedLists

    - How to create an empty LinkedList in Java.
    - same as above (how do all the List operations work)

- What addFirst(), getFirst(), removeFirst(), addLast(), getLast(), removeLast() do.

- How to write a method/program segment that uses all of the LinkedList methods.

- How to **efficiently(!)** iterate through a LinkedList performing some operation.

- Our Own Implementation

    - How do we implement a LinkedList?

    - Know how to write code to add/remove an element from the head/tail of a singly or doubly linked list.

    - Know how to write code to loop through a singly or doubly linked list printing out all of the nodes.

    - What is the running time of all of the List operations using a linked list implementation?

5. Analysis of Algorithms

    - Know the definitions of:

        - Running time (time complexity)

        - Space complexity

        - algorithm

        - Correctness of an algorithm

• Know how to analyze the running time of a simple algorithm and be able to explain why that is the running time.

• Know how linear and binary search work and which is better in which situations.

6. Stacks

- Know how to use a Stack based on our implementation
- Know what the Stack operations do.
- Be able to solve a problem using a Stack.

7. Queues

• Know how we implemented our own queue.
• Know how we specified our own Queue interface.
• know what add(), poll(), peek(), isEmpty(), and size() do.

8. Deque

- How to use the Java Deque interface to write code that would use stacks or queues
- That the LinkedList class implements the Deque interface.

7.