CSC 360 Final Exam study guide
Date for the final: Wednesday, December 10, 10:10 am – 12:10 pm

This will be a closed-note, closed-book, in-class exam. You will write your answers on paper (non-electronic exam). The exam will be 2 hours long. The exam will consist of writing code, examining code and describing what it does, what it will output, what is wrong with the code, or how to make changes to the given code, and questions on concepts. The exam covers the following material/chapters.

Chapter 21:
Know what generics are and why you might use them. Know the diamond notation and how to declare and instantiate a Generic object. Know how to define your own Generic classes and then how to create an object of a specific type, such as creating MyGenericClass<T> and then MyGenericClass<String> foo=… Know how to bound a Generic class using extends on the type such as <T extends Numeric> and <T extends Comparable>. Know how to implement Comparable on a Generic class. Know how to implement a Generic method and the role of the wildcard to indicate that the parameter type must be bound by the given type (e.g., <? extends class>, you will not have to know <? super class>). Know what type erasure is. Know these restrictions on Generics: you cannot create a Generic array, you cannot use a primitive type when declaring/instantiating a Generic class.

Chapter 22/23:
Understand the concept of a data structure and an ADT. Know the basic idea behind a Collection, List, Queue and Set. Specifically, know which of these concrete classes you would use for different situations: PriorityQueue, LinkedList, ArrayList, Stack, HashSet, LinkedHashSet, TreeSet, Map, HashMap, TreeMap and LinkedHashMap. Know what an Iterator is and its methods next, hasNext, remove and be able to use the for-each (iterator for loop). Know how you can take one Collection object and instantiate a different one (e.g., take a LinkedList l and create =new ArrayList(l);) Be able to use the Collections class to take a List object and sort it, shuffle it, obtain the largest and smallest values and obtain the frequency of a particular object. Know the following methods for Stack: empty, peek, pop, push. Know the following methods for LinkedList: empty, offer, poll, remove, peek, element. Know the following methods for a set: isAnElementOf, intersection, union, difference, subset, superset. Know what a hashing function is and the basic idea behind hashing. Know how hashing can improve access efficiency and when its performance degrades (collisions). Know the difference between a list, a queue and a stack in terms of where you can access, insert and remove. Know the difference in outputting a HashSet, a LinkedHashSet and a TreeSet.

Lists:
Know the differences between an ordered and unordered list and the advantages and disadvantages. Understand the difference in each one's efficiency (you do not need to know the Big-O notation, just be able to compare them). Know how to implement an ordered and an unordered list using a linked list implementation.

Threads:
Know what a thread is. Know how to declare a Thread from an instance of a Runnable class. Know how to implement Runnable. Know the Thread methods of setPriority, join, sleep. Know how Threads can cause data corruption and the solution using the synchronization modifier on a method. Know what producers and consumers are and the problems of a shared buffer.