CSC/CPE 103: Fundamentals of Computer Science III  
Fall 2014  

Midterm 1 Study Sheet

Date: Friday, October 24, 2014.  
Time: section 01/02: you will take the test during your lab hour (9\textsuperscript{10} - 10), in room 20-139. Lab session will be held in 8\textsuperscript{10} - 9 a.m. time slot in your regular lab room.  
section 03/04: you will take the test during your lecture hour (10\textsuperscript{10} -11) in your regular lecture room. The lab session will follow as usual.

Type: closed book exam.

Below see the topics covered in this exam. For each topic I mentioned the chapter and the section in the textbook where you can find the related material. I recommend that you take as a guide the lecture handouts I provided for you since in the book there are certain things that we covered a little differently (you are expected to know the material the way it is covered in the classroom). Note that any part of the course material (covered in lectures, labs and projects) can be included in the test, and there won’t be anything in the test that is not related to that material.

Topics:

1. Algorithm Analysis (chapter 2):  
   - Know general rules for computing the time function of different code segments. Be able to work out the exact formula representing the time function, as well as give the big-Oh estimate for the time function of a given code segment.

2. Analysis of List operations (Lists are in Chapter 3, sections 3.2-3.5 – we just need analysis).  
   - Be able to give the big-Oh estimate for the time function of every basic operation in the List ADT (insert, delete, search, check if empty) for both array and linked list implementation.

3. Stacks and Stack operations (Chapter 3, section 3.6):  
   - Know what a stack is and how it can be implemented. Know stack operations and how they work; be able to illustrate on examples. Know both implementations of stacks – array and dynamic linked list implementations. In both implementations be able to write code for stack operations, as well as for additional operations related/applicable to such structures. Be able to write a client code to use stacks.  
   - Be able to give and explain the running time estimates for different stack operations (for both, array and linked list implementations).

4. Queues and Queue operations (Chapter 3, section 3.7):  
   - Know what a queue is and how it can be implemented. Know queue operations and how they work; be able to illustrate on examples. Know both implementations of queues – circular array and dynamic linked list implementations. In both implementations be able to write code for queue operations, as well as for additional operations related/applicable to such structures. Be able to write a client code to use queues.  
   - Be able to give and explain the running time estimates for different queue operations (for both, array and linked list implementations).
5. Priority Queues and Heaps (Chapter 6, sections 6.1-6.3 including)
   - Know what a priority queue is and how it can be implemented (several ways). Be able to give running time estimates for different implementations.
   - Know what a heap is and what the heap properties are. Know different properties of complete binary trees: the index relationship between the parent and its children, the relationship between the number of nodes and the height of the tree, the relationship between the level-number and the number of nodes at that level, the relationship between the total number of nodes and the number of leaves/non-leaves in the tree.
   - Know how a heap is usually implemented (the array implementation). Know the basic heap operations and how they work; be able to illustrate on examples. Be able to write code to implement heap operations, as well as additional operations related/applicable to such structures. Be able to write client code to use a binary heap.
   - Be able to give and explain the running time estimates for different heap operations.

**Lecture handouts related to the material covered in Midterm 1:**

- Data Structures and Abstract Data Types
- Stack ADT
- Queue ADT
- Rules for Exceptions and Exception Handling in Our Programs
- Introduction to Algorithm Analysis
- Computing the Time function and finding its tightest upper bound
- Computing the Time function and finding its tightest upper bound (MORE examples)
- Analyzing Stack, Queue, and List Operations
- Priority Queues and Heaps
- Priority Queues and Heaps (continued)
- Generic code for Comparable Objects

**Supplementary handouts containing sample codes** for the parts of following assignments:

- Lab1 (main)
- Lab2 (main)
- Lab3 (driver and LQueue)
- Lab4 (AStack) + Project1 (MyStack)
- Lab6 (AQueue)
- Lab7 (TreeWork)