CSCI 204 Final Exam Study Guide
Fall 2017

1. **The Rules**
    - The official length of the exam is three hours. I am aiming for about 1.5 hours, though you may use up to three hours if necessary.
    - Get permission from the dean if you need to take the exam in any time other than specified.
    - You may bring the review sheets of paper with anything you like hand-written on it, one or two sided. No printed information is allowed. **Put your name on it**. These review sheets will be returned.
    - Any code or information on this sheet which comes from the Internet, a non-CSCI 204-course textbook, or another person needs a citation. Information from the Internet or from an outside book or person may only be used on this exam with citation and at the instructor's discretion. Always ask if you are unsure.
    - Do not bring any electronic devices such as calculator, cell phone, tablet, or laptop. If you have to have one with you (such as a cell phone), turn the sound off and don't take it out of your bag.

2. **Exam Topics**

The exam is comprehensive. It will cover all topics in the entire semester, though the topics after the second exam will be more than its proportion of length (class days). Listed below are the topics we covered since our second exam. See the study guides for the two mid-term exams for other topics.

A successful CSCI 204 student should be able to
- **In the area of sorting**
1. Explain how various sorting algorithms work, including quicksort, heapsort, and mergesort;
2. Implement the above listed sorting algorithms;
3. Show the time complexity of the above listed sorting algorithm.

- **In the area of hashing,**
4. Explain the concept of hashing;
5. Explain how conflicts may be resolved in hash tables;
6. Define and use various hashing functions;
7. Define and use functions that resolve conflicts, including linear probe, quadratic probe, secondary hashing;
8. Implement all methods for HashMap ADT;

9. **General format of the exam**

The exam will have some short conceptual questions you'd answer or explain; some code comprehension where a segment of code is given and you explain the meaning and output of the code; and some Python functions or classes you'd write.