

CSCI 204 Exam 2 Study Guide

Fall 2017

1. The Rules

- You may bring one sheet of paper with anything you like hand-written on it, one or two sided. No printed information is allowed. Put your name on it.
- 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.
- You hand in your helper sheet with your exam. I'll give it back later. I reserve the right not to grade your exam if you use but forget to include your helper sheet. If you don't use one, please indicate so on the cover sheet of the exam.
- The exam will begin at the hour and promptly end at the 52 minute mark. If I make an exam too long, I will deal with it later.
- Do not discuss the exam with anyone else except the instructor from the time it starts until after the exam is over. Do not discuss the exam with students not in our section before or after the exam until all exams are graded.

2. Exam Topics

Below is the list of major student learning outcomes for this portion of the class. The exam will be based on this list.

A successful CSCI 204 student should be able to

- **In the area of stacks,**
 1. Explain the concept of a stack;
 2. Implement stacks using various data structures such as a Python list, an array, or a linked list;
 3. Write application programs using a stack, for example, converting infix expression to postfix expression, evaluating postfix expressions, determining if a string is a palindrome, finding a path within a connected graph (depth-first search), among others;
 4. Explain and properly use the relation between a stack and recursion.
- **In the area queues,**
 5. Explain the concept of a queue;
 6. Implement queues using various data structures such as a Python list, an array, or a linked list;
 7. Implement bread-first search using a queue;
 8. Use queues to implement various applications;
 9. Explain the concept of a priority queue;
 10. Implement priority queues using various data structures such as a Python list or a linked list;
 11. Implement a circular queue using either a linked list or an array;
 12. Explain the difference and commonality between clock-driven simulation and even-driven simulation.
- **In the area of binary trees,**
 13. Explain what is a binary tree;
 14. Explain various properties of a binary tree;

15. Implement a binary tree using linked nodes or an array;
16. Implement and use various applications using a binary tree, including expression trees;
17. Explain and implement heap using an array;
18. Use heap for sorting;
19. Explain and implement binary search trees;
20. Use BST to search an item, to find maximum and minimum, and to sort in ascending or descending order;
21. Explain the concept of balanced search tree.

3. 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.