

CSCI1321 Final Review

This final will be formatted in a manner very similar to the midterm. There will be 10 questions though you have a longer period of time to answer them. During the exam you will be able to have the Java API open on your computer, but no other references will be allowed.

Topics:

- Graphics – You need to understand how to add custom graphics to a Java application. If I give you a piece of code, you should be able to tell me what it does. I could also ask you to write a short piece of code that performs a certain task.
- Linked list based stacks, queues and priority queues – You learned about stacks and queues in the first half of the semester using arrays. You need to know how they are implemented with linked lists as well and how the two implementations compare. Be able to write or trace such code. Also, you need to understand how a sorted linked list can be used to implement a priority queue and be able to write or trace it.
- Exceptions – You must understand the concept of exceptions and how they work in Java (this includes both syntax and semantics). You should also be able to compare and contrast exceptions to other forms of error reporting. Know about the different types of exceptions and what they are used for.
- Recursion – Know how to write and trace recursive functions. You should also understand when recursive functions can/should be used to aid in problem solving. How do they compare to loops and when are they superior to loops?
- Binary Trees – Know what a binary tree is and the things we can do with them. This includes binary search trees and the different traversals. You also need to understand what they are good for, their O speed, and when that breaks down. You need to be able to write basic functions to do things like find the height of a tree or count the number of leaves.
- Heaps – You need to understand how a binary heap is implemented with a complete binary tree that has “heap ordering”. You should also know how to emulate a complete tree through the use of an array. Be able to write or trace code for this.
- Threads – You need to understand the basic concept of what a thread is, how we create a thread in Java, and the basics of the `java.util.concurrent` package.
- Files and Streams – Know how the stream library is set up in `java.io` and how we can use it for things like file access, binary data, and object serialization.
- Networking – You need to have a basic understanding of how we do communication between computers in Java. You should also understand how this fits in with streams. Also you need to know the material covered on RMI for doing networking where you don't have to deal directly with the socketing.