

CPSC 131 Midterm Exam Study Guide

Fall 2020

When

- 131-06: Wednesday, October 21 during normal class time

Logistics

- You MUST be logged into Zoom, with working microphone and camera. No exceptions. You will be monitored and recorded during the exam.
- You must show a picture ID before taking the exam.
 - Turn OFF your Zoom background and hold your ID up to your camera for at least 5 seconds. You may then turn on your Zoom background.

Format

- Closed-books, closed-notes. You may access only the Canvas Quiz. Absolutely no internet surfing, no collaboration in any form with anyone else.
- On-line Canvas Quiz resembling that of our checkpoints
- There are four parts to the exam, each weighted roughly the same
 - Multiple choice, true/false, short answer
 - Data structure sketching
 - Analyzing alternatives essay
 - Coding
 - You may be asked to add and implement a new member function to one of our existing abstract data type classes taken from our Implementation Examples.
 - Your implementation may require you to add an overloaded helper function as well.
 - Your code must be syntactically and semantically correct. Approach this as though you need to compile and run the code you write.

Topics Covered

- Part 0 - Introduction & Review
 - C++ pointers, references, arrays, dynamic memory, Object Oriented Programming (OOP), classes/structs, templates, exceptions
 - Writing/updating classes with proper

- encapsulation (public/private)
 - instance attribute data members
 - constructors, destructors
 - overloaded queries, accessors, and mutators
 - overloaded operators
 - etc.
- Algorithm complexity analysis
 - asymptotic analysis (big-O)
 - efficiency classes for the fundamental operations of all the data structures covered
 - $O(1)$, $O(\log_2 n)$, $O(n)$, and $O(n^2)$
 - choosing a container by comparing efficiency classes of operations
- Memory model (stack vs. heap)
- Iterative and recursive algorithms
- Part 1 - Sequence Containers
 - Arrays & Vectors
 - Fixed, bounded, & unbounded implementations
 - Amortized efficiency, complexity analysis
 - Lists
 - Singly & doubly linked lists
 - Null-terminated, two dummy nodes, circular with one dummy node
 - Complexity analysis
 - Concepts & Interfaces
- Part 2 – Iterators
 - Iterator Concepts & Interfaces
 - Pointers as iterators
 - Container Traversal Techniques
 - Iterative & recursive
- Part 3 - Container Adapters
 - Stacks, Queues
 - Concepts & Interfaces
 - Array, Vector, List implementations

- Fixed sized arrays
- Bounded and unbounded vectors
- Singly and double linked lists
- Complexity analysis