CMSC 202 Final ExamTopic Outline

Also see Midterm 1 & 2 topics lists!

Textbook Chapters and Sections:

Chapter 15
Chapter 16
Chapter 17 (Section 3)
Chapter 18
Chapter 19 (Sections 1 and 2; Section 3, subsection *Running times and Big-O Notation*)

Topic Outline

Polymorphism

Static vs. Dynamic Binding; equivalently, Early vs. Late Binding

What is polymorphism good for?

Use of the "virtual" keyword

Pure virtual functions and abstract classes; use and syntax

Polymorphic functions; how they work; why they're useful; syntax

Virtual destructors; when to use them and why; syntax

Templates

Benefits of templates

Templated functions; syntax

Which classes can be used with a templated function, e.g. templated sort requires a class that defines comparison operators (> and/or <)

Templated classes; syntax. Examples: templated linked list, templated sort

Compiling templated classes

Writing a container class

Iterators and the STL

- Some STL containers: vector, list, pair, map; should be especially familiar with vector

- Iterators; why are they needed with STL containers?

- Iterator syntax; relationship to pointer syntax; use of iterators in for-loops

- How to write an iterator class; how to write begin() and end() functions in a container class

Miscellaneous

- Memory leaks and valgrind

    - What causes memory leaks? How do you fix them?

    - How does valgrind help you detect memory leaks?

- Exceptions

    - What are the advantages of exceptions over other methods of error handling?

    - When is the use of exceptions appropriate?

    - When and how do you use throw, whether with a basic exception type or an exception class?

    - How do you define an exception class?

    - When and how do you use catch? What is the purpose of catch(…)?

- Recursion and Complexity

    - Big-O notation; linear vs. quadratic run-time

    - Big-O running times of linked list traversal and linear search.