

CMSC 202 Final Exam Topic Outline

Also see Midterm 1 & 2 topics lists!

Textbook Chapters and Sections:

Chapter 15

Chapter 16

Chapter 19 (Sections 1 and 2)

Chapter 19 (Section 3, subsection *Running times and Big-O Notation*)

Chapter 13 (Sections 2 and 3; Section 1 may be helpful for background)

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

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

Types of iterators: forward, reverse, const, bi-directional

Recursion and Complexity

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

Big-O running times of linked list traversal vs. *cached* linked list traversal (e.g. Project 5)

Simple recursive functions that return a value, e.g. power(x, n) or fib(n)

Binary search on a sorted array; compare efficiency to simple linear search

Miscellaneous

C++11 Features

The "auto" keyword; when is this useful?

Range-based for loops; how do they simplify code?

What is multithreading? Why is it useful?

What are smart pointers? Why are they useful?

Memory leaks and valgrind

What causes memory leaks? How do you fix them?

How does valgrind help you detect memory leaks?