Final Exam Study Guide

The final exam is cumulative and covers all of your required reading from AoP, but focuses on the C++ chapters. However, while you need to be an expert in all of the material for your professional life, we will not test the material in chapters 27–29 this semester. This study guide indicates at what level you should know each topic.

For those topics you have the most practice with, listed in *Writing code*, you should be able to write correct, leak-free code. For those topics you should know well enough to execute by hand and draw pictures of, listed in *Reading code*, you should be able to execute the code by hand and determine the output. For those topics you read and understood, but did not practice writing code for, listed in *Concepts*, you should be able to answer conceptual questions, and if the item has a **P** by it, you should be able to draw a conceptual picture of what is happening. Of course, you should also be able to read code with and answer conceptual questions about any of the topics in writing code. Similarly, you should be able to answer conceptual questions about any of the topics in reading code.

Writing code

- Devising an algorithm
- Basic C++ concepts including references, strings, IO, and `const`
- Dynamic memory allocation
- Constructors, destructors, rule of three
- Templates
- Common classes in the STL, *e.g.*, `std::map`, `std::set`, `std::vector`, `std::list`[1]
- Inheritance
- Declaring an exception class, throwing an exception, writing a try/catch block
- Linked lists
- Binary search trees
- Hash tables

[1] Write your favorite methods in your notebook. You might want to make a note of how to use their iterators as well.

Reading code

- Valgrind errors
- Dynamic and static dispatch
- Propagation and handling of exceptions

Concepts

- Data structure efficiency (big O)
- ADTs
- (**P**) Min and max heaps
- (**P**) Priority queues
- (**P**) Graphs
- Sorting

Test format

The exam will consist of:

- Multiple choice and fill in the blank questions emphasizing conceptual knowledge (no partial credit)
- Manipulation of data structures: applying specified changes to data structures and performing traversals to show how the structure changed (no partial credit)[2]
- Reading code problems (partial credit)
- Writing code problems (partial credit)

You will be allowed to bring one notebook, which must contain only things handwritten by you, on which you can write anything you like. You are permitted one tab with *All of Programming* open. The process and proctoring will be like the midterm: you must wait to be checked in, where the TA will verify that you are sharing your desktop and have only two tabs open: the exam in Sakai and *All of Programming.* A TA will visit several times during the exam and make sure you are abiding by the honor code. When you are finished, you should "Ask for help" and wait for a TA to come and verify your submission. Make sure you submit your exam *before* the timer ends, or Sakai may lose some of your work on the current page.

The exam will have 100 points, and you should be fluent enough in the material that you can finish it in 100 minutes. However, you will have 120 minutes to complete the exam to allow for technical issues.

[2] In a pencil/paper exam, we would ask you to draw the resulting data structure