

Study Guide to the Final Exam

Exam Topics

You are responsible for the topics covered in the whole course. Emphasis will be on the material since the midterm: Lecture notes 13–25, the accompanying class demos, and the concepts used in homework assignments 3–8.

Not everything in the lecture notes was covered in class, but you should read any slides that were skipped over to make sure you have a general understanding of what they are about.

Index to the Lecture Notes

Below is a list of all sections and subsections from the lecture notes along with links to the files where they appear. You can use this as a high-level overview of the course.

| | | |
|----|-------------------------------------|------------------------------|
| 1 | About This Course | [lecture 01] |
| 2 | Topics to be Covered | [lecture 01] |
| 3 | Kinds of Programming | [lecture 01] |
| 4 | Why C++? | [lecture 01] |
| 5 | C++ Programming Standards | [lecture 01] |
| 6 | Task List | [lecture 02] |
| 7 | C++ Overview | [lecture 02] |
| | 7.1 C++ Language Design Goals | [lecture 02] |
| | 7.2 Comparison of C and C++ | [lecture 02] |
| 8 | Building a Project | [lecture 02] |
| | 8.1 C/C++ Compilation Model | [lecture 02] |
| | 8.2 Project management | [lecture 02] |
| | 8.3 A sample project | [lecture 02] |
| 9 | Integrated Development Environments | [lecture 02] |
| 10 | Submission Instructions | [lecture 02] |
| 11 | Insertion Sort Example | [lecture 03] |
| | 11.1 Program specification | [lecture 03] |
| | 11.2 Monolithic solution | [lecture 03] |
| | 11.3 Modular solution in C | [lecture 03] |
| | 11.4 Modular solution in C++ | [lecture 03] |
| 12 | Classes | [lecture 03] |
| | 12.1 Header file | [lecture 03] |
| | 12.2 Implementation file | [lecture 03] |

| | | |
|------|---|--------------|
| 12.3 | Main program | [lecture 03] |
| 12.4 | Building <code>InsertionSortCpp</code> | [lecture 03] |
| 13 | Problem Set 2 Preview | [lecture 04] |
| 14 | C++ I/O | [lecture 04] |
| 15 | End of File and I/O Errors | [lecture 04] |
| 16 | Functions and Methods | [lecture 05] |
| 16.1 | Parameters | [lecture 05] |
| 16.2 | Choosing Parameter Types | [lecture 05] |
| 16.3 | The Implicit Argument | [lecture 05] |
| 17 | Derivation | [lecture 05] |
| 18 | Objects of Class Types | [lecture 05] |
| 19 | Construction, Initialization, and Destruction | [lecture 06] |
| 20 | Reference Types | [lecture 06] |
| 21 | Reference Types (cont.) | [lecture 07] |
| 22 | Etudes in Coding | [lecture 08] |
| 23 | Problem Set 1 Design Issues | [lecture 08] |
| 24 | Brackets Example | [lecture 08] |
| 25 | Following Specifications | [lecture 09] |
| 26 | Bytes and Characters | [lecture 09] |
| 27 | Overview of PS3 | [lecture 09] |
| 28 | Brackets Example (continued from lecture 8) | [lecture 10] |
| 28.1 | Stack class | [lecture 10] |
| 28.2 | Brackets class | [lecture 10] |
| 28.3 | Main file | [lecture 10] |
| 29 | Storage Management | [lecture 10] |
| 30 | Copying and Assignment | [lecture 11] |
| 31 | Custody of Objects | [lecture 11] |
| 32 | Move Semantics | [lecture 11] |
| 33 | Uses of Pointers | [lecture 12] |
| 34 | Feedback on Programming Style | [lecture 12] |
| 35 | Bar Graph Demo | [lecture 13] |
| 36 | Handling Circularly Dependent Classes | [lecture 14] |
| 37 | Modeling the Think-A-Dot Machine | [lecture 14] |
| 38 | Polymorphic Derivation | [lecture 15] |
| 39 | Uses of Polymorphism | [lecture 15] |
| 40 | Introduction to the C++ Standard Library | [lecture 15] |
| 41 | Remarks on Upcoming Assignment PS5 | [lecture 16] |

| | | |
|----|--|------------------------------|
| 42 | Remarks on PS4-Consensus | [lecture 16] |
| 43 | Standard Template Class <code>vector<T></code> | [lecture 16] |
| 44 | Overview of PS5 | [lecture 17] |
| 45 | Move Demo | [lecture 17] |
| 46 | Bells and Whistles | [lecture 17] |
| 47 | The Many Uses of Classes | [lecture 18] |
| 48 | Virtue Demo | [lecture 18] |
| 49 | Linear Data Structure Demo | [lecture 18] |
| 50 | Functions Revisited | [lecture 18] |
| 51 | Operator Extensions | [lecture 18] |
| 52 | Exceptions | [lecture 19] |
| 53 | Throwing an Exception | [lecture 19] |
| 54 | Catching an Exception | [lecture 19] |
| 55 | Rethrowing Exceptions | [lecture 20] |
| 56 | Uncaught Exceptions | [lecture 20] |
| 57 | Singleton Design Pattern | [lecture 20] |
| 58 | Smart Pointer Demo | [lecture 20] |
| 59 | Singleton Design Pattern (revisited) | [lecture 21] |
| 60 | More on Functions | [lecture 21] |
| 61 | Casts and Conversions | [lecture 21] |
| 62 | Templates | [lecture 22] |
| 63 | PS6: Who prints the blockchain? | [lecture 23] |
| 64 | STL Iterators | [lecture 23] |
| 65 | STL Algorithms | [lecture 23] |
| 66 | Name Visibility | [lecture 23] |
| 67 | Remarks about PS6 | [lecture 24] |
| 68 | Clocks and Time Measurement | [lecture 24] |
| 69 | Demo: Stopwatch | [lecture 24] |
| 70 | General OO Principles | [lecture 25] |
| 71 | Function-Like Constructs | [lecture 25] |
| 72 | Design Patterns | [lecture 25] |