

- Exam 2 will focus mainly on chapters 10-16, and the questions will be meant to test ideas in those chapters. However, it will be assumed that you are familiar with material from chapters 1-9, also, since some of those concepts might be used in asking the questions.
- Questions will not necessarily be evenly spread over the chapters. For instance, there's likely to be less from Chapter 15, but a whole lot more from Chapter 14.
- The plan is for it to again be 60 multiple choice questions (bring a scantron). Like last exam, about half of them will involve code-like examples, and about half will be more general questions.
- I will still have questions where you need to choose the BEST answer, so please be sure to read all answers and choose the one that is most correct.
- The questions will be designed mainly to test concepts. I will not try to put in obscure syntax questions or things that could easily be mistaken for a typo. If you see those cases, please check with me during the exam. For instance, on the last exam, I accidentally capitalized "if" to "If" – that was just a typo error, not an intentional mistake. I might accidentally leave off a semicolon or something. If you are wondering whether a problem is intentional or not, just ask.
- The test is not meant to cover C++ Syntax, but you will need to know syntax to answer questions. You should not need to focus on things that are library specific
- Here are specific topics that you should plan to study. This is **NOT** necessarily everything you need to know for the test, but it will be a pretty good coverage
 - Covering the Review questions and the Terms from each chapter should be a minimum starting point. I won't repeat those here, but you should be familiar with them.
 - Also, review the topics covered in the slides in class. Generally, the most important topics will be covered in slides, and sometimes the slides will go into more detail (e.g. on event driven programming) than is in the textbook.
 - From Chapter 10:
 - File structure, opening/reading/writing a file
 - Error handling in I/O operations – types of errors, how they can be checked, how they can be dealt with
 - Overloading input and output streams to I/O new data types
 - From Chapter 11:
 - Formatting output
 - File positioning, binary files
 - String streams as a way of writing to a string
 - Generally, reading and processing input streams
 - From Chapter 12:
 - Hierarchies of interfaces (building libraries on top of libraries)
 - The general window/display/shape organization
 - General use of attaching graphics to a window to get rendering
 - From Chapter 13:
 - Combining elements to achieve an overall goal
 - Basic shapes and how you would use them, given a header
 - Note: You do not need to memorize the structure of calls to shape commands.
 - Note: You **do** need to understand how to use them if you were given access to a header file.
 - From Chapter 14 (The most important chapter on this exam!):
 - Object oriented design: encapsulation, inheritance, and polymorphism – what are they and how do they differ?
 - Encapsulation: what is it, why is it used, how is this reflected in code
 - Inheritance: when you would use it, why, how it works
 - Polymorphism: what is it, why is it useful, how does it work in code
 - Make sure you understand things like abstract classes, virtual functions, derived classes, etc.

- My goal is not to test your knowledge of C++ syntax, but I will have to use C++ syntax to test your knowledge of the underlying principles!
 - So, if you study any code specifically, make sure you understand the way that encapsulation, inheritance, and polymorphism are reflected in code!
- From Chapter 15:
 - General idea of graphing data
 - From a function
 - From a file
- From Chapter 16:
 - Event-driven programming: organization of code, use of callbacks,
 - GUI interfaces
 - Getting info from/sending info to a screen
- Other general concepts:
 - Given a header file, you should be able to understand how to use the functions described in that header file.
 - Design a class hierarchy, knowing where different functions should go