

Study Guide for Midterm Exams 1 and 2

You are allowed one 8 ½” x 11” piece of paper with your notes on one side. Each exam day you can use different notes. The notes must be **handwritten** and **your own work**. No photocopies or electronic or typed documents are allowed.

Textbook: Chapters 7 to 11 (review of basics), Chapters 12 to 14 new material. Exam will include C and C++ syntax. You must know the syntax of each two languages do something different given the same syntax.

You can access the text in Canvas.

Best wishes for success.

Course Objectives:

1. Demonstrate solid intermediate-level competence in problem analysis and program construction in C and C++.
2. Demonstrate, an introductory level, competence in program design according to object-oriented principles.
3. Demonstrate skill and fluency in program testing.
4. Use pointers, strings, arrays, structures, and classes appropriately in programs.
7. Use text files and streams for data input and output.

Topics

- **Chapters 7 through 11** - review of C and introduction of C++ syntax. This material should be well understood and you should know it fluidly. I will assume that you know the basic concepts and syntax in all questions, this includes:
 - Numeric data – all types
 - Characters type and related functions
 - Arrays – declaration and use
 - Pointers – declaration and use, esp. with functions.
 - Strings – differences between C and C++, declaration and use.
 - Be able to draw and explain data diagrams for all of the above types
 - Math, character, and string libraries. Know the functions and features we have used.
 - Loops: while, for, do..while, infinite for loop, case. Be able to design programs with any type of loop. Be able to discern the best choice to accomplish a task and explain your reasoning. Be able to analyze code with loops and explain what is happening.
 - Be able to trace processing in a loop or any program and identify the values of all variables at all times (given the relevant data).

Study Guide for Midterm Exams 1 and 2

Topics

- **Chapter 12 and 17.1 and 17.2:** Strings in C and in C++.
 - The string library.
 - String functions.
 - You should be able to answer any questions on how to read strings, write strings and format strings in both C and C++.
 - Differences between C and C++ representation of strings.
 - Especially where the differences will trip you up (where the same statement will give very different results.)
 - Fundamentals of vector template class's data members, functionality, and usage.
 - Using the Vector template class such as declaring a vector of any data type, and using the basic functionality such as .size() and .capacity().
 - Using the methods we discussed, especially when adding to a vector using .push_back(), know what happens when another item is added when the vector is full.
 - Using the other vector methods we discussed. Not iterators.
 - Memory management of vectors and difference between a vector and a simple array.
- **Chapter 14:** Files and Streams in C and C++.
 - Be able to describe, use and explain the purpose of all of the predeclared streams in C and C++. Be able to explain which are buffered and unbuffered and why.
 - Be able to use a stream to reference a file for input or output in text form in C and C++.
 - Be able to use a stream write to a file in text form in C and C++.
 - Be able to use a stream to read from a file in text form in C and C++.
 - Be able to detect and gracefully handle common file I/O problems.

Study Guide for Midterm Exams 1 and 2

Topics

- **Chapter 13 Enumerated types and Structures.**
 - Be able to define or discuss enumerated types in C and C++.
 - Be able to declare the type, a variable of the type and use an enum in C or C++.
 - Be able to use an enum with an array to generate messages based on the enum types in C++.
 - Be able to describe, design and write code and explain how to use the enumerated type in a program. Be able to justify your choice.
 - Be able to draw and explain data diagram of an enumerated type.
 - Be able to define or discuss structures in C and C++.
 - Be able to use a struct in C or C++ to create and use a variable of that type.
 - Be able to create and assign a pointer to point at a structure.
 - Be able to describe, declare and use an array of structures.
 - Be able to describe, write code and explain how to directly access data members of a structure. Dot operator.
 - Be able to describe, write code and explain how to indirectly access data members of a structure using a pointer. Arrow operator.
 - Be able to draw and explain data diagram of a structure or an array of structures. With and without a pointer to the structure.
- **Chapter 13 Classes.**
 - Be able to relate a structure to a class in C++, be able to describe what is the same, what is different. Be able to discuss the reasons why a class should be used in C++ over a structure. Describe the design advantages of a class for Object Oriented Design.
 - Be able to write a simple header and .cpp file for a class in C++.
 - Be able to include a class in another program and use it correctly to create a single instance of the class or an array of objects of that class.
 - Be able to reference the data members and functions of a class in your program or in function definitions of the class using an object of the class.
 - Be able to reference data members and functions of a class in your program or in function definitions using a pointer at an object of the class.
 - Be able to discuss the implied parameter in any context.

Study Guide for Midterm Exams 1 and 2

Topics

- **Programming Performance Project in C++**
- You will be asked to write a program in class on day two of the exam using your laptop or a school PC with Eclipse IDE or Xcode. If you plan to use a school PC, be familiar with whatever IDE is on it, most likely Xcode on the Mac partition. On your own device, you can use the IDE of these two that you prefer.
- You may reference a single side of one page of notes and your own programs.
- You will **Not** be able to use the book, my slides, my sample programs, language documentation of any form or any other source of help.
- You will need to show me that you can:
 - Write a program that reads from a text file and console input, and can output to the console and a text file.
 - Use an array of some simple or complex type.
 - Use an enumerated type.
 - Use strings.
 - Bonus for being able to write a Class.
 - Everything I've shown you and discussed in C++ is fair game.
- Your performance will be observed and evaluated throughout the programming process. Assessment will include your problem-solving approach, debugging skill, and familiarity with the language.