CS 2212/6610 Intermediate C/C++

Study Guide for Final Exam

You are allowed **one 8 ½" x 11" piece of paper with your notes** on both sides for this exam. Your own code can be written on the notes if you like. **Your notes must be your own work and handwritten or they will be taken away. You are not allowed any other source of help. This is the link to the Final Exam Schedule in *my*Charger.**

This is an on-ground course, students mast take the exam in person with instructor and TA proctoring. The final exam is written (paper and pen). Bring a calculator with no connectivity (not a phone/tablet/IoT device). Students who are ill or quarantined will be allowed to take the exam by arrangement only if the proper documentation is provided to the University. A grade of INC might be assigned depending on the timing of these events.
Best wishes for success. LP

Textbook: The **final exam is cumulative**, but the questions are emphasizing the latter part of the course. Chapters 7 to 11 (review of C/C++ basics), and Chapters 12 to 14 were on the Midterm, this material is mostly in support of your ability to write more sophisticated programs. Final Focuses on: Chapters 12 to 20. The text is in Canvas and you are expected to read the chapters and use it to study.

The final Exam will focus on the material covered since the first exam. The recommended text has reference material on most of this content. Occasionally, some material will not be covered before this exam (Recursion for example) and will not be in the exam. An alternative assessment will be used for this material at the end of the term in that case.

Course Objectives and Assessment

1. Demonstrate solid intermediate-level competence in problem analysis and program construction in C++. *This is primarily assessed through your Programming Projects and general final exam questions on where things go, how to put things together address this objective.*
2. Demonstrate, at an introductory level, competence in program design according to object-oriented principles. *P2 and final exam questions on classes and design.*
3. Demonstrate skill and fluency in program testing. *This is primarily assessed through your Programming Projects including PF, general final exam questions on testing.*
4. Use pointers, strings, arrays, structures, and classes appropriately in programs. *This is primarily assessed through your Programming Projects. Final exam questions asking you to write short pieces of code or interpret what a piece of code does.*
5. Manage dynamic allocation appropriately in C++. *This is primarily assessed through your Programming Projects, and final exam questions Vector class and its use.*
6. Apply bit-level operators in low-level applications. *Bitwise operation program and questions on your final exam. Shift, bitwise &, | and ^, masking with bitwise operations. Hexadecimal to binary conversion.*
7. Use text files and streams for data input and output. *This is primarily assessed through your Programming Projects, final exam questions on writing and reading from a file.*

DISCLAIMER: This is not intended to be a completely comprehensive list, or to act as a substitute for your own notes and reading. Any topic in the assigned reading, notes or discussed in class may also be on the test, even if it does not appear on this review sheet.

CS 2212/6610 Intermediate C/C++

Study Guide for Final Exam

8. Simulate execution of common algorithms for arrays. *This was a part of your programming projects, and there will be final exam questions on these algorithms and asking you to write code for this.*
9. Create projects with more than one compile module. *This was a requirement of P2, there will be final exam questions on design and asking you to write code related to projects with more than one compile module.*

Topics:

1. Ch 11 and throughout the rest of the book: Pointers (Objective All, esp. 4 and 10)
    a. Use with strings
    b. Use with arrays
    c. Use with structures and classes
    d. Use with functions
2. Ch 12 Strings (Objective 1, 3, 4)
    a. String functions and processing
    b. Methods of accessing subsets or a single character in a string.
    c. Using stringstream to redefine the contents of a string as a stream - see read().
3. Ch. 14 - Streams and Files (Objective1, 3, 7)
    a. Declaration of a file stream for I/O
    b. Strings as streams
    c. Reading and writing to streams and files.
    d. Especially text files form any data type.

Exam 1 Stopped Here.

4. Ch 13 - Enumerated types (Objective 1, 3, 4)
    a. use of enum in code
    b. array of coordinated messages
5. Ch 13 and 20

Structures: (Objective 1, 3, 4)

    a. Declare and use a structure.
    b. Create a structure with data
    c. Update data in a structure
    d. Arrays of structures

Classes: (Objective 1, 2 3, 4 and 10)

    e. Object Oriented principles related to class design.
    f. Responsibility to create common built-in class functions for comparison and Input/output
    g. Declaring a class with a header and .cpp code file.
    h. Including a class in your project
    i. Create an object of a class with or without data
    j. Constructors and destructors
    k. Create an array of a class or a vector of a class
    l. Use as Base Type (BT) for vector or arrays of class objects.
    m. Writing and using class functions
    n. Pointers to objects or structures (examples with functions)

DISCLAIMER: This is not intended to be a completely comprehensive list, or to act as a substitute for your own notes and reading. Any topic in the assigned reading, notes or discussed in class may also be on the test, even if it does not appear on this review sheet.

CS 2212/6610 Intermediate C/C++

Study Guide for Final Exam

Topics:

6. Ch. 16 & 17- Dynamic allocation, Vectors and Iterators: (Objective 1, 3, 5 and 10)
    a. growth by reallocation - Flex Array
    b. new, delete, and delete[]
    c. Vector class – *very important*
    d. Using functions of the Vector class such as .push_back()
    e. Iterators
7. Ch. 17 and 18 - Arrays and 2D arrays: (Objective 1, 3, 4, 5, 6 and 9)
    a. matrices – using myArray[][]
    b. arrays of pointers to objects
    c. using pointers to process arrays
    d. reference them from functions using pointers or by passing address
        i. Examples: comparison functions for quick sort, binary search, etc.
8. Ch. 15 - Bitwise Operations: (Objective 1, 3, and 6)
    a. Binary and hexadecimal notation and fluid conversion between them
    b. Integral variable representation for signed and unsigned integral variables
    c. covering bit operators: right and left shift, &, |, ^ are emphasized.
    d. Using masks with operators and the function of each operator with a mask
    e. Algorithms discussed: encryption, reversal of bits, and parity.
9. Ch. 20 – Making Programs General (Objective 1, 2, 3, 4, 9 and 10)
    Command-line arguments:
    a. What change is required to main() to accept passed parameters, argc and argv[]
    b. How to use argc and argv[]
    c. What information is always passed by the operating system?
    d. Be able to send arguments from IDE or command line to your program
    e. Be able to read and process arguments in your program.
    Functions:
    a. Functions with generic type
    b. Functions as parameters.
10. Ch 19 - Recursion: (objective 1, 3, 8)
    a. Recursion on arrays
    b. Examples: math operations: factorial, Fibonacci series, binary search, quick sort, etc.

DISCLAIMER: This is not intended to be a completely comprehensive list, or to act as a substitute for your own notes and reading. Any topic in the assigned reading, notes or discussed in class may also be on the test, even if it does not appear on this review sheet.