

The Final examination will be Thursday, May 8, from 3 - 4:50 PM in ITT 322. It will be closed book and notes, except for **three** 8.5" x 11" sheets of paper (front and back) with notes. The test will be comprehensive, but focus on material covered since test 2. New topics covered since Test 2 are:

Chapter 9. Pointers

- Declaration of a pointer variable using '*', e.g., "int * myPtr;"
- Address of operator (&) and indirection operator (*)
- Relationship between arrays and pointers
- Pointer arithmetic, "walking" pointers down an array, comparing pointers
- Pointers as function parameters (relationship between pointer parameters and pass-by-reference), returning pointers from functions
- Dynamic memory allocation, null pointer

Chapter 10. Characters, Strings and the string Class

- General idea of the C++ library functions for characters (ctype): isalpha, toupper, etc.
- General idea of the C++ library functions for C-Strings (cstring): strlen, strcpy, etc.
- C++ string class: definition of string objects, supported string operators (>>, <<, =, etc.), and string class member functions (append, length, c_str, find, replace, etc.), getline

Chapter 11. Structured Data (skip section 11.11 on Union)

- Definition of a structure (struct)
- Accessing structure members using the dot operator (.)
- Array of structures
- Structures as function arguments and returning a structure from a function
- Pointers to structures ("->" operator)
- Enumerated data types

Chapter 12. Advanced File Operations

- General usage of istream and ostream objects to read/write to files

Chapter 13. Introduction to Classes

- Procedural vs. Object-oriented programming
- class definition, access specifiers (private, public), inline functions, constructors, destructor
- creating objects/instances of a class
- pointers to objects and usage of (->) to reference class member functions
- arrays of objects
- Die class example(s)

Chapter 14. More about Classes

- static vs. instance variables
- operator=, deep vs. shallow copies, copy constructor
- operator overloading, this pointer
- friend function: operators: << and >>
- aggregation - a class contains an instance of another class

STL (from class not in text) -- Try get the general concepts and be able to read code using STLs

- Standard Template Library (STL) container classes: vector, deque, list, set, multiset, map, multimap
- STL iterators and their usage
- STL algorithms

Recursion (from class not in text)

- General idea of recursion (be able to trace recursion using the run-time stack)
- Recursive examples to "study": countDown, power, factorial, fibonacci, binary search, merge sort
- Problem of solving fibonacci recursively via recursive divide-and-conquer algorithm and the general solution using dynamic programming