Final Exam Review

CS 1428
Fall 2017

Jill Seaman

Final Exam

- 006 (9:30 class): Thurs 12/14, 8:00am to 10:30
- 007 (12:30 class): Thurs 12/14, 11:00am to 1:30
- In this classroom, closed book/notes, clean desk
- Comprehensive (covers entire course)
- 30% of your final grade
- Bring your ID card and a number 2 pencil!!!
- NO: calculators or cell phones.
- NO: headphones/earbuds.

Exam Format

- 150 Points total
  - 70 points: 35 multiple choice and T/F (scantron form)
  - 80 points: writing code on the test paper
    - ➡ programs, functions and individual statements
- Tasks:
  - Tracing code (what is the output, etc.)
  - Finding errors in code
  - Evaluating C++ expressions
  - Demonstrate general knowledge about C++ and programming
  - Programming (writing code)

Content from Textbook

- 1.1-3
- 2.1-17 (except 2.11)
- 3.1-10
- 4.1-15 (except 4.13)
- 5.1-12
- 6.1-5, 7-10, and 13
- 7.1-4, 6, and 8
- 11.2-8

See reading list online for specific topics of each section

Ch 1: Intro to Computer and Programming

- Definitions: Computer, Program, Programmer
- Hardware vs Software
- Hardware components: (cpu, main memory, secondary storage, input and output devices)
- Program vs. Algorithm
- Programming languages: machine lang vs low level lang vs high level lang
- Compilation: source code file -> executable
- Execution

Ch 2: Introduction to C++

- cout and << (output)
- Literals: numbers, characters, strings
- Identifiers, rules for valid names
- Variable Definitions and Initialization
- Assignment Statements
- Data Types
  - int, short, long, float, double, bool, char, string
- Scope rules, comments, named constants

Ch 3: Expressions and I/O

- cin and >> (input)
- Numerical Expressions: precedence rules
  - Operators: +, -, *, /, % (modulus)
- Type Conversions: implicit and explicit
- Integer division vs float division
- Multiple/combined assignment
- Pow(a,b) and other Math library functions
- Formatted output: setw, setprecision, fixed
- Inputting characters and string (>> vs getline)

Ch 4: Making Decisions

- Relational and Logical Expressions
  - Rel. Operators: <   <=   >   >=   ==   !=
  - Logical Operators: !   && ||
- Decision statements:
  - if and if-else
  - nested if statements and if-else if
  - block or compound statement
  - **switch**
- Scope of variables in blocks

Ch 5: Loops and file i/o

- increment/decrement operators (`x++, x--`)
- while loop (general purpose)
- do-while (body done at least once)
- for loop (init; test; update)
- Which loops are good for which situations
- Count controlled, sentinel controlled loops
- Keeping a running total, input validation
- Nested loops, infinite loops
- break and continue
- File I/O: filestream objects, reading/writing

Ch 6: Functions

- Function definition (implementation in code)
- Function call (void vs one that returns a value)
- Function prototype, when it is required
- Function parameters and arguments
- Passing arguments by value and by reference
- Return statement
- Returning values from functions
- Scope: variables, local vs global, lifetime

Ch 7: Arrays

- Array declaration/definition, size is constant
- Array elements, syntax, range of subscripts
- Array initialization: `int list[] = {6,7,8};`
- Processing arrays
  - input, output, sum, average, finding max, min
  - counting values that pass a test, array assignment (copy)
- Lack of bounds checking
- Functions and arrays

Ch 11: Structures

- Structure Definition (with members)
- Defining structure variables (having struct type)
- Struct var initialization: `Student s1={“Bob”,3.2};`
- Accessing members (dot operator)
- Operations over structures
  - assignment, function call
  - input/output, comparison (define yourself)
- Arrays of structure, processing them
- Nested structures
- Structures and functions

Software Development Process

- Top Down Design
  - Break tasks into subtasks
  - Make a hierarchy of tasks
- Incremental Development
  - Implement one piece at a time
- Testing
  - Test cases: input values and expected output
- Debugging
  - Strategy: output values of variables
  - Strategy: output literals to trace execution path

Sample problems

See the lecture entitled:

Final Exam Exercises

on the website

How to study

- Review the slides (Units 1-7)
  - understand all the concepts, **quiz yourself**
- Use the book to help understand the slides
- Review programming assignments (fix yours!)
  - get printouts of solutions Thursday OR during office hours
- Review the previous exams
- Do the Final Exam Review Exercises (slides)
- Practice, practice, practice! Write code!
- Get some sleep

Office hours

Day	Date	Time
M	12/4	3:30-4:30pm
T	12/5	2:30-3:30pm
W	12/6	3:30-4:30pm
Th	12/7	3:30-4:30pm
T	12/12	2:30-3:30pm
W	12/13	11:00am-12:30pm
F	12/15	3:30pm-4:30pm

*and by appointment