

# Final Exam Review

## Contents

1	A Major Portion of the Exam	1
2	Stuff from the Midterm and Earlier	2
3	New Stuff...	2
3.1	Functions . . . . .	2
3.2	Arrays & Vectors . . . . .	3
3.3	Pointers . . . . .	3

---

## 1 A Major Portion of the Exam

- We've talked about many things in this class, but from the beginning I've stressed what the goal is. . .
  - Who can remember?
- 

- There's an old saying in computer science. . .

*Ideal software is Designed by Philosophers. . .*

*Implemented by Mathematicians. . .*

*and Tested by Idiots.*

- There's no shortage of the last category, but the first two are in short supply
- 
- *Understanding problems. . .*
  - *Figuring out how to solve problems. . .*
  - and *Designing solutions to problems* are the focus of any software engineering course and at the heart of computer science
  - We've covered these things in detail, and there will be questions on the final related to this
  - These questions will account for a little less than *half* the total of the final exam

- There *will* be a design question like on the midterm, which will require more time than any other question
- 

## 2 Stuff from the Midterm and Earlier

- Since the final is cumulative, *everything we've covered is a possible question*
  - Here's a list of some things you may want to read up on
    1. **C++ Operators and Order of Evaluation.** This is something that you simply have to memorize. It was stressed repeatedly up until (and including) the midterm. There will be questions related to this (similar to what you've had) on the final.
    2. **C++ Data Types.** You should know the basic data types, and their advantages/disadvantages are.
    3. **Casting.** This is something that was not covered on the midterm, or any quiz. People repeatedly have problems on homeworks related to this, however. There's a good chance some questions regarding casting will be on the final (e.g. Will C++ be able to cast from *this* type to *that* type? If so, will C++ give me a warning? If it does give me a warning, *why* do I get such a warning?)
    4. **Conditional Control Structures.** What are these... what do they accomplish. (Everyone should know these by now!)
    5. **Looping Structures.** Name all three... what can you say about each? What are each best suited for?
    6. **Code Comprehension related to the above.** You should know how to recognize all of the above in example C++ code.
    7. **MOST IMPORTANT:** Know how to write your name at the top-right of a piece of paper!
- 

## 3 New Stuff...

- *Most* of the exam (excluding the design related questions), will cover everything from chapter 6 onwards
  - I'll now outline what you should know (warning: this is essentially everything, but I'll stress some things along the way)
- 

### 3.1 Functions

- Functions are critical. I can honestly say that a *major* portion of your exam will require you to be proficient with functions and *everything* covered in chapter 6 *and* in class
- You should know all key phrases that I've quizzed you on (twice, since almost everyone got them wrong on the first quiz)
- You should know what the *storage modifiers* are, and what they accomplish
- Scope rules will be tested heavily (§6.10 covers this *extensively*)
- Parameters, passing schemes (value/reference/pointer), overloading, and templates are all things you should know

- Recursion... know what it is, how to identify, and *maybe* how to write a sample recursive function (or two)
  - You should be able to comprehend code using functions
- 

### 3.2 Arrays & Vectors

- You should know what an array is, how it's allocated, what one can do with arrays and *how* that works
  - You should know the various ways in which arrays are declared
  - You should know some of the ways in which arrays can be passed to functions, why you would choose one method over the other, etc.
  - Know the basics of multidimensional arrays
  - You should also know *some* way of searching arrays... a *simple* search will do
  - Also, everything above only relative to *vectors*
  - You should be able to comprehend code using arrays & vectors
- 

### 3.3 Pointers

- There will be pointer related questions and code-comprehension questions
- The code comprehension questions, however, *will not be too tricky* (it's too easy to make pointer questions that even Deitel would require 10 minutes to figure out... I will not do this)
- Declaring and initializing pointers, pointer operators, the variations of `const` pointers
- You should *really know* the relationship between pointers and arrays... this *will* pop up
- Function pointers may be on the exam... but only the most abstract level of understanding would be required (just make sure you read §8.12)