SOME REVIEW

Course summary

This course introduced the *science* of computing. We learned tools and techniques affecting all parts of the software lifecycle.

Specifications:

- Writing precise specifications
- Reading precise specifications

Design:

- Looking at a problem abstractly.
- Standard abstractions that have proven useful in computer science — abstract data types.
  - list
  - stack
  - queue
  - tree

Summary (cont'd)

- Some new data structures that offer alternative ways to implement an abstract idea.
  - linked lists
  - circular queues
  - trees
  - binary search trees

  You'll see more of these in CSC 263.

- Analyzing the efficiency of algorithms
  - Big-O notation
  - Reasoning about non-recursive programs

Summary (cont'd)

Implementation

- Properties of a good program.
  - abstraction
  - reuse
  - information-hiding
  - encapsulation
  - design by contract, documentation
- Designing a program to have these properties.
  - interfaces
  - abstract classes
  - exception handling
  - Java memory model
  - Java-specific concepts: `Iterators`, `Comparable`

Summary (cont'd)

- New algorithms
  - merge sort
- A new programming technique: recursion
  - thinking recursively
  - termination, base case
  - recursion vs induction
  - recursion vs iteration

Summary (cont'd)

Code Reviews

Testing

- Choosing a systematic and thorough set of test cases.
- Documenting testing so that it will be convincing.

Documentation

- Design by contract
- Internal comments
  - representation invariants
  - how much to comment and where
- external comments
  - pre- and post-conditions

Final exam

- 3 hours, no aids allowed
- Material from
  - lectures
  - labs
  - assignments

Some possible exam questions

- trace code/test code
- write iterative/recursive code
- write code that deals with trees, linked lists, arrays, stacks, etc.
- do big-O analysis of code
- talk about pros and cons of various implementations for an ADT
- design/implement/use a class
- implement an iterator
- talk about any of the above things
- correct mistakes in examples of any of the above things

How to study for the exam

- Go back over anything you never felt good about; ask us about it
- Practice! Do the sample exams on the web. Both old midterms and finals will be helpful.
- Go to office hours, even if you do not have specific questions.
- We will be holding additional office hours - watch the announcements on the course web page.

How to write the exam

- Read over the whole exam before you start.
- Budget your time. You could run out of time if you lose track of time (even though the exam isn't intended to be a speed test).
- Read questions with care, and be sure to do what we asked for!
- If you are at all unsure about what we mean, feel free to ask a question.

End-of-term administrative stuff

- Marks:
  - Watch the web page for announcements when various marks have been posted.
  - Check your marks!
  - You are responsible for ensuring that we have correct marks for you
  - Report errors immediately, by bringing the relevant assignment or test to your instructor.

    Please do this BEFORE the final!

What is ahead?

- System support
  - Operating systems
  - Compilers
  - Programming languages and paradigms
  - Databases
- Communications and security
  - Network protocols
  - Security
  - Concurrent systems (how to write them? how to test them? how to reason about them?)

What is ahead (cont'd)?

- Theory
  - Which things can be computed and which cannot?
  - How to analyze code?
    - Solving recurrence relations
    - Proving algorithms correct
    - Analyzing running times
    - Time/space trade-off
  - New data structures
  - Logic (the centre of it all)
- Creating usable and scalable systems
  - Human-computer interaction
  - Requirements analysis
  - Design patterns, architecture, modeling languages
  - Metrics to access/predict cost, quality, complexity
  - Simulations

What is ahead (cont'd)?

- And much much more...
  - Artificial intelligence (robotics, planning, vision)
  - Graphics
  - Numerical analysis

And in conclusion...

Hope you enjoyed this class!

Hope you will choose computer science as your future career!

And good luck on the final exam!