COMP 250

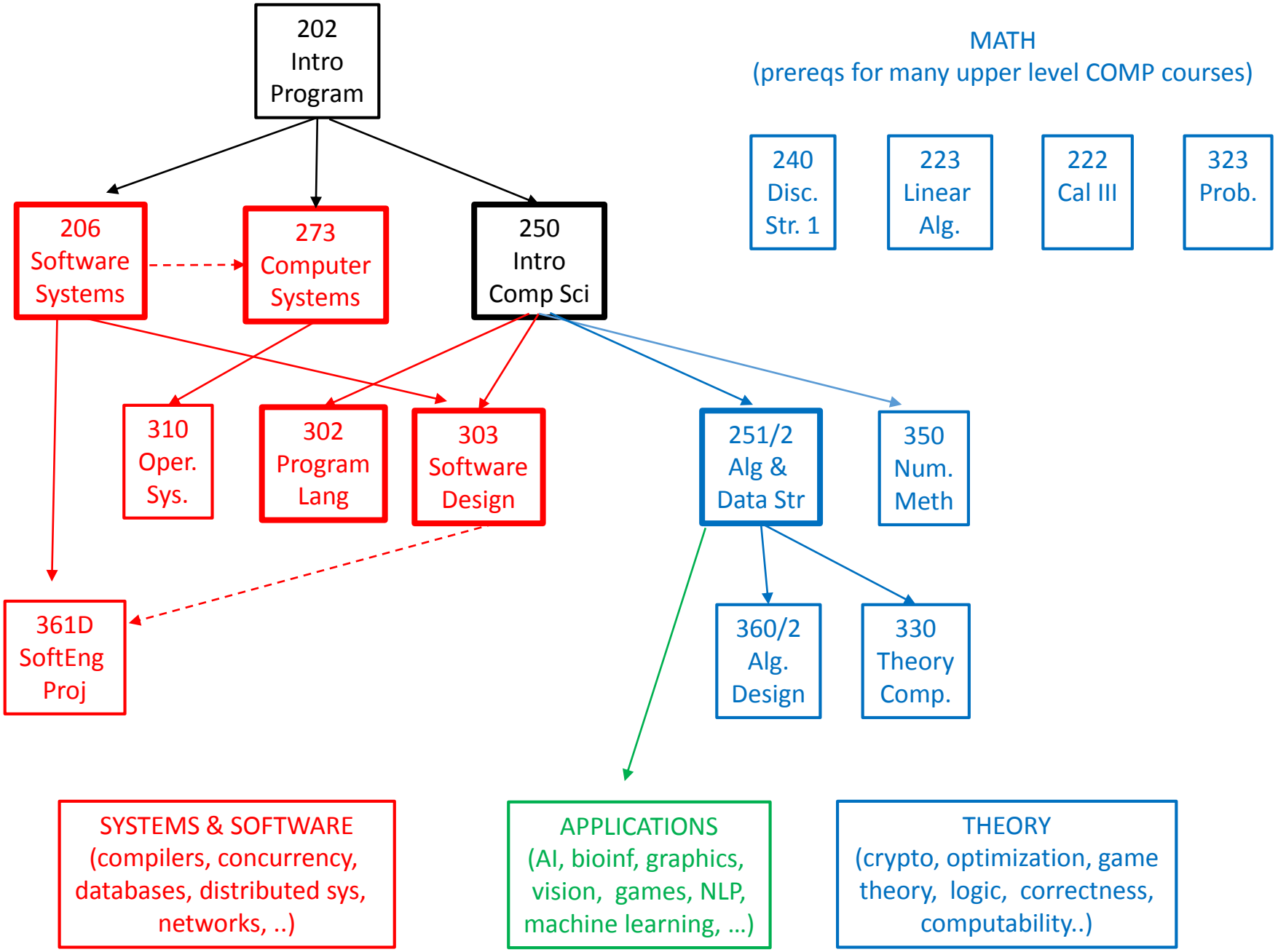
Lecture 36

MISC

- beyond COMP 250

- final exam comments

Dec 5, 2016


202
Intro
Program
MATH
(prereqs for many upper level COMP courses)
240
Disc.
Str. 1
223
Linear
Alg.
222
Cal III
323
Prob.
206
Software
Systems
273
Computer
Systems
250
Intro
Comp Sci
310
Oper.
Sys.
302
Program
Lang
303
Software
Design
251/2
Alg &
Data Str
350
Num.
Meth
361D
SoftEng
Proj
360/2
Alg.
Design
330
Theory
Comp.
SYSTEMS & SOFTWARE
(compilers, concurrency,
databases, distributed sys,
networks, ..)
APPLICATIONS
(AI, bioinf, graphics,
vision, games, NLP,
machine learning, …)
THEORY
(crypto, optimization, game
theory, logic, correctness,
computability..)

COMP 251 Algorithms & Data Structures

http://www.cs.mcgill.ca/~jeromew/comp251.html

http://www.cim.mcgill.ca/~langer/251.html

for exercises and links to MOOC videos….

MATH 240 or MATH 235 are strongly recommended.

(As of Fall 2017, they will be official co-requisites for COMP 251.)

In Winter 2017, assignments will be a mix of Java programming and theory (calculations, and proofs).

COMP 252 (Honours version of 251)

http://luc.devroye.org/252.html

MATH 240 required.

Lectures and assignments are theory only. (Only pseudocode.)

Very challenging and interesting course. Instructor is fantastic.

(To prepare, see MOOC links from my COMP 251 page.)

“Systems”

- Multiple interacting parts that create a complex whole

- Multiple levels too (“low”, “high”)

COMP 206 Intro to Software Systems

http://www.cs.mcgill.ca/~jvybihal/Courses/CS206/index.html

Unix and bash scripting, C & python, networks & web

Course Description:

(since ~2000 *and now out of date*): Comprehensive overview of programming in C, use of system calls and libraries, debugging and testing of code; use of development tools like make and version control systems.

(starting 2017): Programming of software systems: system calls and libraries, manipulation of binary data and dynamic memory, network communications, foreign function interfaces. Associated tools for build configuration, debugging, and version control.

COMP 273 Intro to Computer Systems

http://www.cs.mcgill.ca/~jvybihal/Courses/CS273/index.html

http://www.cim.mcgill.ca/~langer/273.html (my version)

Topics:

Logic circuits (simulator only, i.e. no hardware)
MIPS processor and assembly language
Memory (cache, virtual memory, disk)
User vs kernel
I/O

COMP 302 Programming Lang & Paradigms

http://www.cs.mcgill.ca/~prakash (also fantastic)

See his ~~propaganda~~ Course Outline from last year:

http://www.cs.mcgill.ca/~langer/250/Prakash-302-outline.pdf

"…we will study … functional programming, imperative programming and object-oriented programming…. The point of this class is not to master the syntax of several different languages but, rather, to learn concepts..."

http://www.cs.mcgill.ca/~langer/250/Prakash-302-schedule.pdf

COMP 303 Software Design

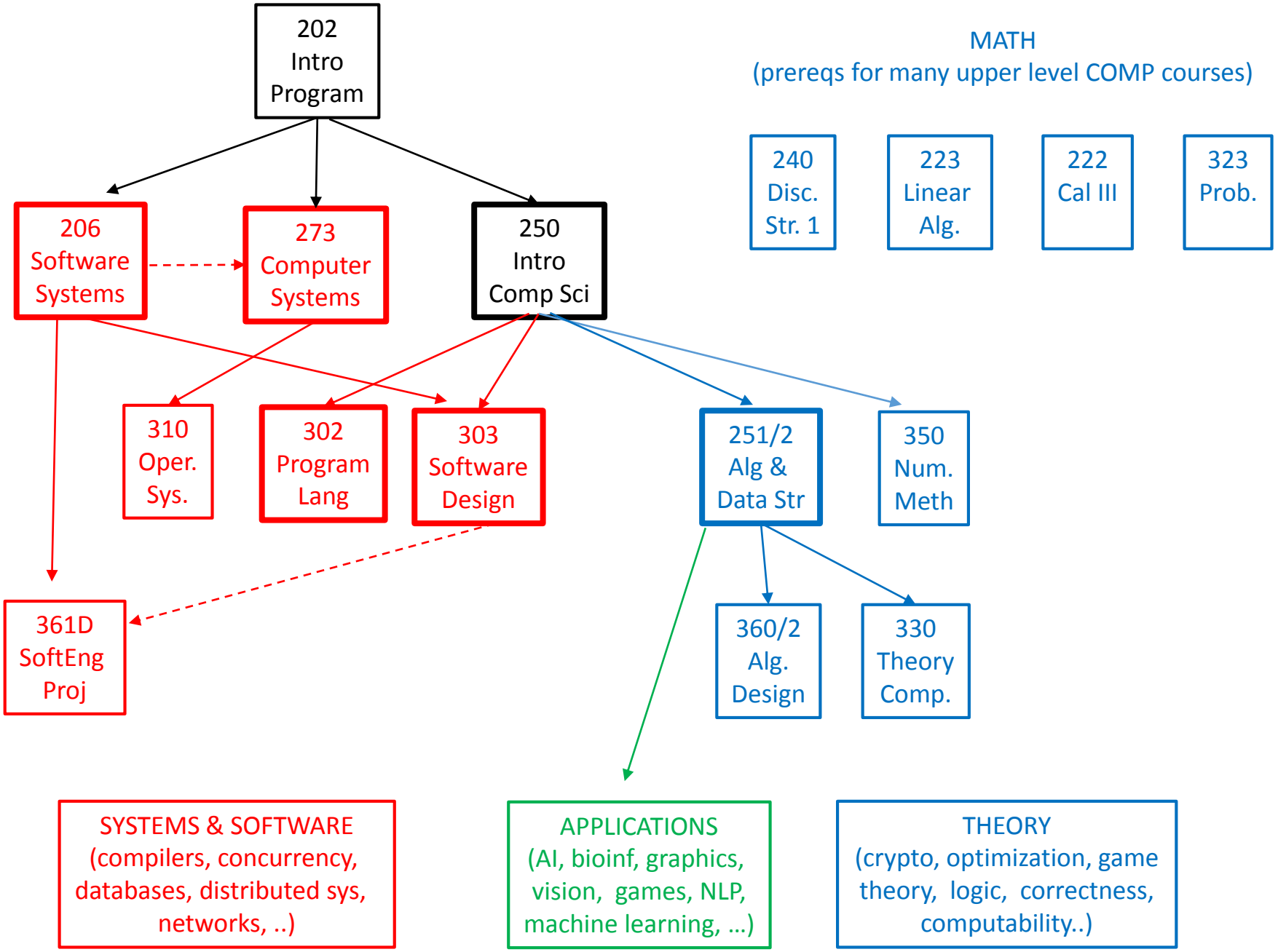
http://cs.mcgill.ca/~martin/teaching/comp303-winter-2016/

From Martin Robillard's page above (Dec. 6):

> "**The official prerequisites (COMP 206 and COMP 250) are the absolute minimum program requirements.** This course targets students who … should be able to, with a minimum of hesitation, write Java programs to solve small and well-defined problems, use a revision control system to organize their work, and use a debugger to trace through execution and inspect run-time values. **COMP 302 and COMP 251 are recommended background.**" *[ML: my boldface added.]*

http://mcgill-csus.github.io/comp_303_feedback.html

There used to be a graded project. Now there is an ungraded project.


202
Intro
Program
MATH
(prereqs for many upper level COMP courses)
240
Disc.
Str. 1
223
Linear
Alg.
222
Cal III
323
Prob.
206
Software
Systems
273
Computer
Systems
250
Intro
Comp Sci
310
Oper.
Sys.
302
Program
Lang
303
Software
Design
251/2
Alg &
Data Str
350
Num.
Meth
361D
SoftEng
Proj
360/2
Alg.
Design
330
Theory
Comp.
SYSTEMS & SOFTWARE
(compilers, concurrency,
databases, distributed sys,
networks, ..)
APPLICATIONS
(AI, bioinf, graphics,
vision, games, NLP,
machine learning, …)
THEORY
(crypto, optimization, game
theory, logic, correctness,
computability..)

COMP programs

Minor (BSc) or Minor Concentration (BA)

- Cal 3 does not belong in that program, in my opinion.

CS Liberal (latecomers, or those needing escape hatch)

CS Major

CS Honours (Required: more theory, MATH)

….

COMP 400: Honours Project in Computer Sciene

As of Fall 2017, the course will be renamed Project in Computer Science. It will be 4 credits and open to anyone (assuming you have supervisor).

Final Exam review

Lectures 1-8 (8 / 40 questions)

Preliminaries

- binary numbers
- grade school alg.

Linear Data Structures: Lists, Stacks, Queues

- data structures
- algorithms: add & remove, sorting

Details you can skip

- Dummy nodes for linked lists
- Stack examples: balancing parentheses, graphics

Possible Questions

- Convert a given number from one base to another
- What is O( ) of a particular operation on some data structure?
- Given a stack or queue, what is the result of applying certain operations?
- Given a list, how does it change after some step of some sorting algorithm?

Lectures 9-16 (8 / 40 questions)

Proofs by Induction

Recursion (examples)

Solving recurrences (back substitution method)

Asymptotic complexity: formal definition of O(), ….

Details you can skip

- induction proof (sketches) that algorithms are correct
- tail recursion
- Claims 3 & 4 about logs (lecture 14)
- subset properties of families of functions O( ), …

Possible Questions

- What's wrong with following induction proof?
- What is base case of some recursive algorithm ? What is the call stack when some condition is met?
- Solve the recurrence …
- Give values for c, n0 in a big O proof.

Lectures 17-28, 34 (17 / 40 questions)

Trees, BSTs, Heaps, Maps, Graphs

- data structures
- algorithms: add & remove, traverse, …

Details you can skip

- expression trees
- using limits to show O()
- worst case analysis of buildHeapFast
- cryptographic hashing and authentication
- ~~tree and graph traversal details using stack and queue~~

  You should know this. See Announcement Dec 8.

Possible Questions

- What is order of vertices visited in some given tree or graph, using some given traversal method ?
- Manipulate a binary search tree or heap by…
- Hash tables: how do they work? What are they good for?

Lectures 29-33 (7 / 40 questions)

Object Oriented Design

- Inheritance and polymorphism

- Object methods (equals, hashCode, toString())

- Modifiers and packages

Details you can skip

- constructor chaining with `super`
- Object.clone()
- `final` modifier

Course Evaluations

- How much work was it ?
- Was it clear to you what you needed to learn and do, and why?
- Was it interesting or useful ?

How much work was it ?

Faculty of Science suggests a 3 credit should require 9 hours work per week for an average McGill student (3.0) to get an average grade (B):

13 weeks * 9 hours/credit/week ≅ 120 hours

What I aim for:

- lectures (40 hours)
- study and exercises (40 hours)
- assignments (40 hours = 4 x 10)

Final Words

- No more official office hours. But I am at university Dec. 5-7, 8, 12-14. Best to email me for appointment.
- Final Exam is Dec 14
- I am out of country Dec 16-22. I hope to post final exam and course grades before Xmas.