COMP2521 19T0
Week 8, Thursday: The Course in Review

Jashank Jeremy
jashank.jeremy@unsw.edu.au

course review
exam information

COMP2521
19T0 lec15
cs2521@
jashankj@
Review

Course Review

COMP2521
19T0 lec15
cs2521@
jashankj@
Review

Course Goals

COMP1511

- gets you thinking like a *programmer*
- solving problems by developing programs
- expressing your ideas in the C language

COMP2521

- gets you thinking like a *computer scientist*
- knowing a set of fundamental techniques/structures
- able to reason about their applicability/effectiveness

Course Goals
(continued)

At the end of this course you should be able to:

- analyse performance characteristics of algorithms (A)
- measure performance behaviour of programs
- choose/develop effective data structures (DS)
- choose/develop algorithms (A) on these data structures (DS)
- reason about effectiveness of data structures + algos
- package a set of DS+A as an abstract data type
- develop and maintain 9999-line C programs

Syllabus Detail
(I)

For each specific data type, we considered:

- implementation in C (data structures, functions)
- operations (e.g. new, insert, delete, search, traverse)
- analysis of efficiency of operations
- applications of the data type

Abstract data types

- interface vs implementation
- defining ADTs in C (`x.h`, `x.c`, `typedef struct x *X`)

Problem-solving approaches

- recursion, divide-and-conquer, generate-and-test

Syllabus Detail
(II)

Sorting methods

- simple sorts: selection, insertion, bubble
- better sorts: mergesort, quicksort
- complexity of various sorting algorithms

Linear structures

- linked-lists: singly linked, doubly linked
- sets, stacks, queues, priority queues, heaps

Trees

- tree terminology, tree properties
- binary search trees, recursive algorithms
- heaps, priority queues

Syllabus Detail (III)

Balanced search trees

- varieties of balanced trees: splay, 2-3-4/red-black, ...
- tree rotations, tree merge, root deletion

Hash tables

- hash functions
- implementation/use of hash tables
- collision handling: chains, linear probe, double hash

Graphs

- graph terminology, graph properties
- implementation of graphs: adjacency matrix, ...
- graph search: depth-first, breadth-first
- minimum spanning tree, shortest path

Final Exam

3-hour exam on Mon 4 February, worth 55% of course mark.

Held in CSE labs (allocations posted on web site soon)

60% based on Practical Part, 40% based on "Theory" Part

Final Exam (continued)

Bring: your student card, a pen, that's all
What's available to you (in the exam and right now):

- online access to Unix Programmers Manual (`man`)
- a C quick-reference sheet (attached to exam)
- The Algorithms Almanack (list of all algos)
- a sheet of paper for rough working (not to be removed)

What you do **not** have access to:

- no access to COMP2521 web site
- no access to your files (labs, assignments, etc.)
- no access to Web, Google, Facebook, Stack Overflow, etc.

Final Exam

the Practical Part (aka “Prac Exam”)

- three small(ish) programming tasks
- aim: check whether you can program in C
- level-of-difficulty: two easy, one not-so-easy
- supplied with test data and `check` script
- once it passes all `check` tests, submit and move on
- partial marks available **if** submitted program compiles
- zero marks if no submission or submission has compile errors
- zero marks for ”table look-up” solutions (extra tests in marking)

Final Exam

the Theory Part (aka “Theory Exam”)

- short-answer questions (about 6, with varying marks)
- aim: check how much you know about course material
- some calculation required; you have on-screen calculator
- cover a wide range of topics from the course
- e.g. what is the output of the above program?
- e.g. what is the depth of the following tree?
- e.g. which edges are in the minimum spanning tree?

Final Exam

(continued)

Some exam strategy tips:

- 180 mins, 90 marks 1 mark 2 mins
- partition time between theory and prac as you like/need
- but don’t spend more than 40 mins on any one Prac question
- if stuck with debugging, work on the next question
- allow at least one hour for theory questions

Revision Strategy

How to revise?

- re-read lecture slides and example programs (see web)
- take a look at old exams
- review tute and lab exercises and assignments
- write some programs
  (programming is a *skill* that improves with practice)

No questions from past exams/labs/assignments will be in the exam.

Supplementary Exams

Supplementary exams are only available to students who

- do *not* attend the exam
- have a serious documented reason for not attending
  (must convincingly show that your ability to study was significantly affected)
- show satisfactory performance in other components of the course

If you attend the final exam

- you are making a statement that you are ”fit and healthy enough”
- it is your only chance to pass (i.e., no second chances)

Supp Exam will be held on Saturday 16 February

- don’t leave the country if you have a Supp and still want to pass

Assessment

Assessment is about determining how well
you understand the syllabus of this course.
If you can’t *demonstrate your understanding*, you don’t pass.
In particular, I don’t pass people just because ...

- please, please, ... my parents will be ashamed of me
- please, please, ... I tried *really hard* in this course
- please, please, ... I’ll be excluded if I fail COMP2521
- please, please, ... this is my final course to graduate
- etc. etc. etc.

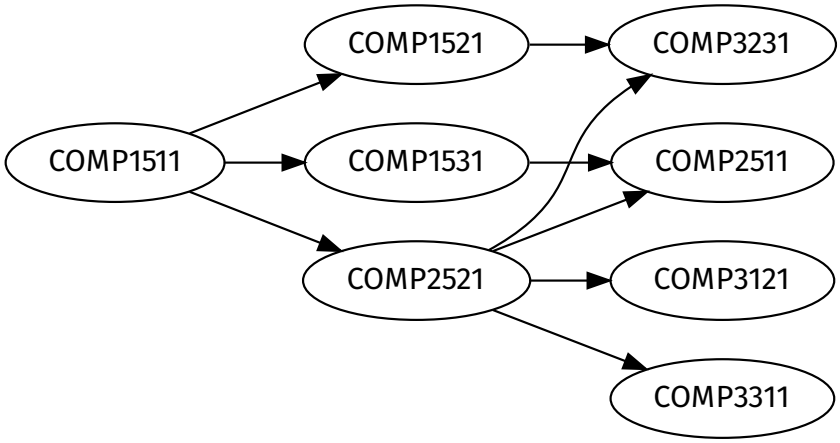
Assessment
(continued)

Of course, assessment isn't a "one-way street" ...

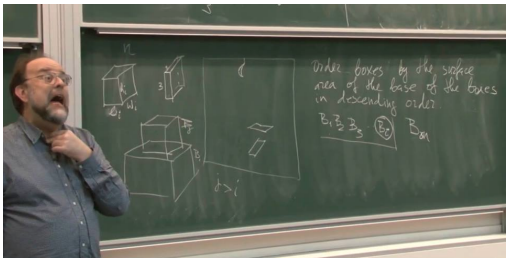
- I get to assess you in the final exam
- you get to assess me in myExperience

myExperience.unsw.edu.au
Telling me good things is fine ...
Telling me things I did wrong is better ...
(If I don't know what's wrong, I don't know what to fix)

Where Next?



Where Next?
The Algorithms Stream



COMP3121/3821 **Algorithms and Programming Techniques** (T1/T2)
dynamic/linear/greedy programming, flow networks, strings, ...

COMP4121 **Advanced and Parallel Algorithms** (T3)
pure theory: PageRank, Markov models, error-correction, ...

COMP4128 **Programming Challenges** (T3)
pure practice: puzzles, challenges, contests; applications!

Where Next?

The Databases Stream

COMP3311 **Database Systems**
COMP9315 **Database Systems Implementation**
COMP9313 **Big Data Management**
COMP9318 **Data Warehousing and Data Mining**
COMP9319 **Web Data Compression and Search**
COMP6714 **Information Retrieval and Web Search**

COMP2521
19T0 lec15
cs2521@
jashankj@
Review

Where Next?

The Formal Methods Stream

COMP2111 **System Modelling and Design**
COMP3141 **Software System Design and Implementation**
COMP3151 **Foundations of Concurrency**
COMP3153 **Algorithmic Verification**
COMP3161 **Concepts of Programming Languages**
COMP4141 **Theory of Computation**
COMP4161 **Advanced Software Verification**
COMP6721 **(In-)Formal Methods: The Lost Art**
COMP6752 **Parameterised and Exact Computation**

COMP2521
19T0 lec15
cs2521@
jashankj@
Review

Where Next?

Other Streams

The Systems Stream

COMP3231/3891 **Operating Systems**
COMP9242 **Advanced Operating Systems**
COMP9243 **Distributed Systems**

The Networks Stream

COMP3331 **Computer Networks**
COMP9332 **Network Routing and Switching**
COMP9334 **Capacity Planning**
COMP9336 **Mobile Networks**
COMP4337 **Securing Wireless Networks**

And Finally ...

good luck with the exam, and with the rest of your computing studies!