

COMP10002

Semester One, 2018

What Happens Now?

Week 12, lecturer: Jianzhong Qi

© 2018 The University of Melbourne

Slides developed by Alistair Moffat

[More Computing?](#)

[Exam Overview](#)

[Genius at Work](#)

[More Computing?](#)

[Exam Overview](#)

[Genius at Work](#)

The most obvious next subject is COMP20007 Design of Algorithms, offered in semester 1 each year.

It is the pathway through to further languages (Java, in SWEN20003) and technologies (databases, in INFO20003).

The pathway through COMP20003 Algorithms and Data Structures (offered semester 2) is also an option, but has some overlap in content with COMP10002.

Subject COMP20005 Engineering Computation is now restricted against COMP10002. It has substantial overlap, and you shouldn't take it.

COMP10001 → COMP10002 → COMP20007 → ...

or

ENGR10003 → COMP20005 → COMP20003 → ...

If you are heading off to a Maths, Physics, Bio, Geomatics, whatever, major, computing skills are still critical. Use your second- and third-year elective slots wisely.

Want to combine two disciplines? Consider adding the Diploma in Informatics, to fit more in to your BSc.

The exam is based primarily around three themes:

- ▶ Writing functions to show your command of control structures in C: selection, iteration, and abstraction
- ▶ Writing declarations and functions to show your command of data structures in C: arrays, structs, linked structures, and combinations of them
- ▶ Demonstrating your understanding of the fundamentals of algorithm design and analysis, including asymptotic execution times under different models, and “good” versus “bad” algorithms

Basic tests:

- ▶ Integer number representations, general idea of floating point representations
- ▶ Simple arithmetic and operations (function to sum a sequence)
- ▶ Loops and selection on scalars (prime numbers and etc)
- ▶ Pointer arguments to functions
- ▶ Recursion.

Basic tests:

- ▶ Arrays, including functions that process arrays
- ▶ Hierarchical construction of types and declarations (using constants, structs, and arrays)
- ▶ Manipulation of structs in functions via pointers
- ▶ Broad principles of recursive structures using pointers: lists and trees.

Basic tests:

- ▶ Functional growth rates and principles of analysis
- ▶ Dictionary data structures, including arrays, lists, and binary search trees; and their tradeoffs and costs
- ▶ Binary and linear search
- ▶ At least one $O(n \log n)$ -time sorting algorithm
- ▶ Problem solving techniques and their various application domains.

Add in:

- ▶ Details of number representations
- ▶ Program correctness arguments, and formulation of loop invariants
- ▶ Pattern matching and string search algorithms
- ▶ Functions over recursive structures (lists and trees)
- ▶ Details of Quicksort, Mergesort, Heapsort, including of their analysis and relative merits
- ▶ Details of all dictionary structures, including hashing
- ▶ Priority queues, including implementation via a heap.

There will be at least one challenge that will involve on-the-spot creativity or insight.

Maybe a problem we haven't discussed, or a related problem that can be solved by adapting something you have been shown for a different purpose.

Maybe an analysis of a new algorithm, or some aspect of something covered only in passing in the lectures.

Maybe to work backward from an analysis, to derive an algorithm that meets it, putting together your knowledge from different parts of the subject.

The sample exam shows these various expectations.

Don't get frightened by “score very well” questions unless you are hoping to score very well!

- ▶ Start well in advance (**now!**)
- ▶ Lay out a study schedule that balances your exam timetable against particular subject needs
- ▶ Put a copy of your exam timetable in a place where others will see it too
- ▶ Scale back your social activities and other commitments
- ▶ Do (some of) your study with other people committed to passing
- ▶ Read the whole book again (or for the first time), and not just the lecture slides

- ▶ Revisit as many of the chapter exercises as you can manage
- ▶ Read the LMS site from top to bottom
- ▶ Review the on-line workshop solutions
- ▶ Do as much programming as you can, writing programs that work.

- ▶ Revisit as many of the chapter exercises as you can manage
- ▶ Read the LMS site from top to bottom
- ▶ Review the on-line workshop solutions
- ▶ Do as much programming as you can, writing programs that work.
- ▶ Eat well, sleep well, and make sure you get regular sunshine and exercise.

- ▶ Highlight the key words in each question. Look out for the words **write a function** (can write more than one)
- ▶ Watch out also for **you may**, **you should**, and **you may not**
- ▶ Draft answers using the blank pages
- ▶ Use parts of one question to help you answer another, especially any program fragments
- ▶ If you cannot write the C code, write answers in English describing the approach you planned to use

- ▶ Be brief and to the point in any written answers
- ▶ Be neat. Use a black or blue pen, or a dark pencil. **No red or green or purple!** And no liquid paper, or pencil rubbings dandruff
- ▶ Ask yourself “what competency is being tested with this question?”. **Then try and deliver**
- ▶ See if you can include a surprise at some stage – an extension, or insightful comment to make the marker smile.

If you think you have found a mistake in the examination paper (it does happen), state what your [assumptions](#) going forward are, and what the [alternative assumptions](#) might have been.

Then [continue on the basis of your assumptions](#).

Don't get agitated about the problem, and don't just freeze and do nothing. We will [assess your response](#) to the unexpected situation, even though that situation may not have been planned.

Hurdles: No one is prevented from sitting the exam. So if you are currently in a perilous state, you should make a considered decision whether to abandon, and focus on other subjects (3/3 is better than 2/4); or whether to try and pull it out of the fire via the exam.

The higher your exam mark, the harder it is for us to fail you, regardless of your project and test marks. Decisions to fail students get made very carefully.

Yes, Assignment 2 marks will be finalised and made available before the exam ([by 10 June](#)).

Yes, there will be consultancy help available during the break: [12pm-1pm Tuesdays 5, 12, and 19 of June, Room 8.14 Doug McDonnell Building](#) (No availability 28 May–3 June).

Yes, I will continue to respond to emails.

What else?

If not already done, please take five minutes to complete the on-line form, linked from the LMS.

The written comments will get read (after the exam results are loaded), and the feedback that is provided will be acted on if it is possible to do so.

But for that to happen, we need you to provide thoughtful feedback. You can see a summary of and response to last year's survey on the LMS.

Finally...

COMP10002

wrapup

[More Computing?](#)

[Exam Overview](#)

[Genius at Work](#)

Good luck with all of your exams...

Finally...

COMP10002

wrapup

[More Computing?](#)

[Exam Overview](#)

[Genius at Work](#)

Good luck with all of your exams...

And remember, Algorithms are Fun!