COMP10001 Foundations of Computing
Final Exam Preparation

Semester 2, 2016
Chris Leckie

October 18, 2016

Lecture Agenda

- Last lecture:
  - Where does computing lead
- This lecture:
  - Final exam preparation

Lecture Outline

1. Preparing for the Final Exam

2. Reflections

Tips for Exam Nirvana

- Get to the exam venue in plenty of time (go the right place at the right time!)
- Put a copy of your exam timetable is somewhere where your family/housemates can see it
- Note that the first 15 minutes of the exam are reading time; **use it wisely**
- Take your student card
- Take writing instruments and not much else
- Apply for special consideration *within 48 hours of the exam via the Student Centre* if you were impeded by illness or other documentable causes

What's Examinable?

- Everything from all the lectures, with the exception of the Advanced Lectures
- *Code/maths* from guest lectures is not examinable, but general concepts **are**
- I won't ask questions that will require you to **write** code that uses knowledge of character encoding
- I won't ask you to write code to generate anything other than the most basic HTML document (e.g. simple list or table)

What's Examinable?

- We don't expect you to write recursive functions from scratch (but maybe fix some recursive code)
- We expect you to be able to use the Iterators described in week 11
- We might ask a question about one of the project exercises
- Topics covered in the mid-semester test are still examinable
- There is a cheatsheet on the LMS under the "MST and Exam" page, but it doesn't summarise everything

What's the Structure of the Exam?

- 2 hours long
- 10 questions
- 120 marks (worth 50% of subject)
- Write everything on the exam paper
- It has the same general structure as the past exam papers on the "MST and Exam" LMS page

The Exam has 3 Parts

- Part 1: Algorithmic thinking
- Part 2: Constructing programs
- Part 3: Conceptual questions
- The marks are *roughly* evenly split between these 3 parts

Part 1: Algorithmic thinking

These are questions that involve tasks such as:

- Evaluating Python "one-liners"
- Tracing the execution of code
- Finding errors in code
- Rewriting a given function in a different way
- Reordering lines of code that are in the wrong order

Week 7 and 8 tutesheets have examples of these types of questions

Part 2: Constructing programs

- Fill in the blanks in a given function
- Write a function that implements a given task
- At least one of these questions will be very difficult!

Part 3: Conceptual questions

- These are short answer questions (no coding)
- Often 1 or 2 sentences is sufficient for each part of the question
- These might ask about topics covered in the guest lectures
- I think these can be the easiest (and fastest) questions to answer, but many students make a mess of these questions

Past Exams

- Several past exams and a practice exam are available on the "MST and Exam" LMS page (with suggested solutions)
- Try these under exam conditions
- Get a friend to mark your answers
- Some of the older exam papers used Python 2 (we use Python 3), but the differences are small

Tips for Revising

- You need to practice writing code on a *blank piece of paper* (there is no Grok in the exam)
- Try writing answers to the old tutorial worksheet questions, Grok worksheet questions, Project questions, examples or class exercises from lectures
- Make a list of places where you get stuck, or errors that you make
- Revise those topics where you keep making errors

Remember

- When we mark your answers, we think like a Python terminal (syntax errors will be penalised)
- Irrelevant or excessively long answers to conceptual questions may be penalised
- We don't mark (or expect) comments in your code

Lecture Outline

1. Preparing for the Final Exam

2. Reflections

How We Ensnared You ...

- Harnessing computation for problem solving
- Fundamental programming constructs
- Data manipulation
- The Web, multimedia and visualisation
- Elements of maths, engineering, logic, design
- Concerned with theories, principles, limits of computation and information
- If you enjoy puzzles, argument, creativity, philosophy and games ... oh and *fun*, you've come to the right place!

We Would Like to Get Your Feedback

- The Subject Experience Survey (SES) is open on-line to students
- Will remain open for the next week
- Go to https://subjecteval.unimelb.edu.au

Final Words

Thanks for giving computing a go!

The COMP10001 team:

Chris	Minh	Angie
Farah	Lachlan	Damian
Andrew	Qingyu	Ned
Kevin	Zahra	Lucy
Milad	Chung Man	Aidan
Yosh	Caiti	Grok Learning

Good luck with your exams and future studies

Computing is fun and creative: it only gets better!