

Foundations of Computing

Welcome and Introduction

Ekaterina Vylomova
Summer Term 2023



THE UNIVERSITY OF
MELBOURNE

January, 17th: On this day...(random facts)

- **1595** – During the French Wars of Religion, Henry IV of France declares war on Spain
- **1773** – Captain James Cook leads the first expedition to sail south of the Antarctic Circle
- **1899** – The United States takes possession of Wake Island in the Pacific Ocean
- **1912** – British polar explorer Captain Robert Falcon Scott reaches the South Pole, one month after Roald Amundsen.
- **1996** – The Czech Republic applies for membership in the European Union

Lecture Agenda

- Who (are the lecturers/tutors/demonstrators)?
- What (is the subject all about)?
- Why Python?
- Where (do I go)?
- How (do I get started)?
- How (does the assessment work)?
- Academic Honesty/Tips?

SpeakUp

- Speakup: <https://web.speakup.info/room/join/45980>
- We'll use SpeakUp for questions, activities during lectures
- Room Code: 45980

Lecture Outline

- 1 Who?
- 2 What?
- 3 Where and When?
- 4 How Do I Get Started?
- 5 Assessment
- 6 Getting Help
- 7 Tips for Winning at COMP10001

Who?

- **Lecturer:**

- Ekaterina Vylomova



Definition

lecturer (n): person who writes/delivers the lectures, coordinates the subject, designs the worksheets/projects, writes the tests/exams, informs, entertains, engages, enthuses, and disentangles the undisentangleable

Who?

- **Lecturer: Ekaterina (Kat) Vylomova**



"I am a Lecturer at the University of Melbourne. I graduated with a BSc, MSc in Computer Science, an extra MSc in Machine Learning, PhD in Natural Language Processing (NLP). Prior to joining academia, I worked as a software developer at various companies specializing on information retrieval, information security, and software virtualization. Now I mainly teach and do research in NLP models for a great variety of world's languages."

Who?

- **Tutors:**

- Amy Mendelsohn (Head Tutor)
- Yige Song (Head Tutor)
- Han Perry
- Andrey Shcherbakov
- Lucy Harrison
- Stanley Wirian
- Patrick Gibbs



Definition

tutor (n): person who runs the tutorial (also called workshops), helps with the marking, provides sagacious advice on subject-related matters, reinforces, empathises, explains, endures (the lecturers), and helps decipher the undecipherable

Who?

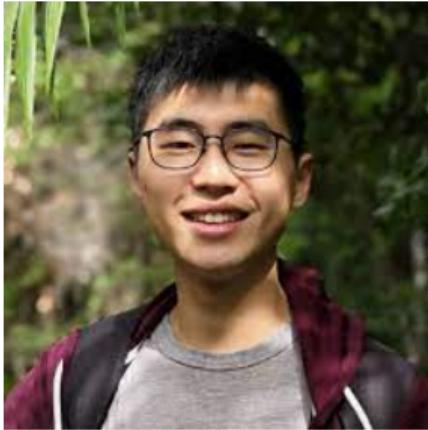
- Head Tutor: Amy Mendelsohn



"Hello! I am a Master of Computer Science student currently working through a one-year research project simulating the movement of pottery in Roman Britain. You can often find me viewing sunsets around Melbourne. I entered computer science in uni through a class similar to this, and hope you enjoy the chase too."

Who?

- Head Tutor: Yige Song



"I am a Master of Computer Science alumni, and will soon start my PhD in AI and education. This is my 4th time teaching COMP10001, and I absolutely love the subject (and Grok). I enjoy travelling, food and reading.

*Dream Destination: Rapa Nui
Favourite Books: (1) Algorithms to Live by; (2) Thinking, Fast and Slow*

Can't wait to meet you all!"

Who?

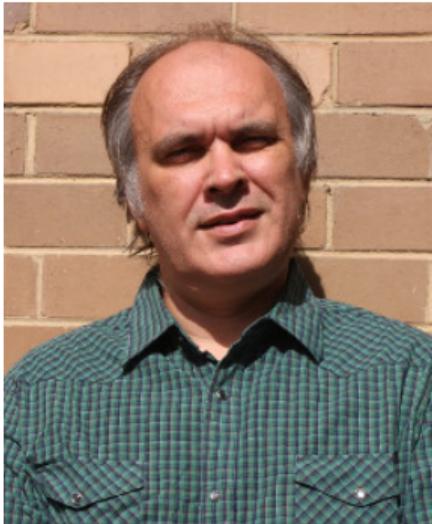
- **Tutor: Han Perry**



"I am Han (they/them). I've just started a Master of Computer Science, after finishing a Bachelor of Science (Maths and Stats) and Diploma of Computing. This will be my 9th semester of teaching this subject! I love computing and programming because there's lots of problem solving, it's so useful to a whole range of applications and also because it feels so satisfying when the code you've been debugging finally works"

Who?

- **Tutor: Andrey Shcherbakov**



"I am an ex-Intel research scientist who recently defended his PhD thesis on formal verification. I previously taught multiple UniMelb courses such as "Web Search and Text Mining", "Stream Computing", "Foundations of Computing", "Introduction into Databases"."

Who?

- **Tutor: Lucy Harrison**



"I am a PhD student in the School of Mathematics and Statistics at the University of Melbourne. My work explores methods of geospatial disease modelling and optimal spatial decision making, applied to zoonotic malaria in Indonesia, and Japanese encephalitis virus in Australia. I love to sew and knit in my spare time!"

Who?

- **Tutor: Stanley Wirian**



"I am a second year Masters of Computer Science student. I've mainly been using Python as a CSIRO Data61 intern, and I hope to share my knowledge in this subject."

Who?

- **Tutor: Patrick Gibbs**



“I am currently completing a Masters research project at the University of Melbourne. Here I am combining my expertise in genetics and computing, developing machine learning models to predict the effect on genetic sequence differences on the phenotype of an organism. My hope is this research will have uses in both agriculture and medicine; allowing farmers to better predict yield, or medical professionals to assess predisposition to disease.”

But More Importantly...Who are You?

- A: Bachelor of Arts
- B: Bachelor of Biomedicine
- C: Bachelor of Commerce
- D: Bachelor of Design
- E: Bachelor of Science
- F: Other

Vote on on SpeakUp: Room 45980

... How much Coding have You done?

- A: Competent user, but never coded or even seen source code
- B: Have seen source code, but that's about it
- C: Have very basic experience with coding, but only enough to know what it is
- D: With lots of looking things up and trial and error, can write basic code
- E: Can code somewhat, but far from fluent
- F: The computer is putty in my coding hands!

Vote on on SpeakUp: Room 45980

Lecture Outline

- 1 Who?
- 2 What?
- 3 Where and When?
- 4 How Do I Get Started?
- 5 Assessment
- 6 Getting Help
- 7 Tips for Winning at COMP10001

What (is the Subject all about)?

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

Programming

- Computer programs are simply sets of steps to complete some task
- Determining what the steps should be requires learning how computers “think” ... and how a particular programming language expresses the way a computer thinks
- At its most basic level, a program is made up of a sequence of **statements** that are executed sequentially one after the other

The Art of Programming

- Syntax
- Algorithms and data structures (D. Knuth, T. Cormen et al.)
- Libraries and frameworks (scipy, pandas, tensorflow)
- Skills and practice ('Code Complete' by Steve McConnell)
- Software design/system architecture (team leads)
- colab tools (git, svn)

The Art of Programming



What is a Computer?

- A big grid/matrix of cells (memory locations)
- Can add, multiply and compare cells really fast (instructions)
- Can run a “program” (list of instructions = machine code)
- At the most basic level, computers use binary (one or zero)
- E.g. to turn top left pixel on the screen red ...

```
10010010 00000001 11111110000000000000000000000000  
10001010 00000001 110100100100101001010100001001
```

Obviously not human friendly

- Easier (assembly language)

```
LDC r1 0xFF000000  
STO r1 #D2529509
```

- Even better (Python-like)

```
screen[0, 0] = (255, 0, 0, 0)
```

- Best?

Hey Sirixa, make the pixel at the top left of the screen red!

There are Lots of Programming Languages

- http://en.wikipedia.org/wiki/List_of_programming_languages
- We will use Python 3.6/.7
- You just write it like a text file, and the Python “interpreter” turns it into machine code for you

There are Lots of Programming Languages



A Message from the Python Creator



“Actually, my initial goal for Python was to serve as a second language for people who were C or C++ programmers, but who had work where writing a C program was just not effective.”
— Guido van Rossum (*the Benevolent Dictator for Life*)

Source(s): goo.gl/mk5ou2

Python

- Easy to learn: interpreted language; interactive experimentation
- Free; open source (python.org)
- Highly readable
- Cross-platform compatible
- Powerful, extensive libraries
- Widely used in industry, science, education, entertainment, ...
- We will use Python v3.6 via Grok Learning

Python

Easy to learn: interpreted language; interactive experimentation

Interpreter: >>>

```
>>> print("Hello World")  
Out: Hello World
```

Python

Easy to learn: interpreted language; interactive experimentation

Interpreter: >>>

```
>>> print("Hello World")  
Out: Hello World
```

Try:

```
>>> import antigravity  
:)
```

Python

Highly readable code

```
>>> print("Hello World")  
Out: Hello World
```

```
>>> 2 ** 100  
Out: 1267650600228229401496703205376  
Operation , not function (pow (2, 100))
```

```
>>> type (2 ** 100)  
Out: <class 'int'>
```

Python

The Zen of Python

```
>>> import this  
Out: The Zen of Python , by Tim Peters
```

Beautiful is better than ugly .
Explicit is better than implicit .
Simple is better than complex .
Complex is better than complicated .
Flat is better than nested .
Sparse is better than dense .
Readability counts . { ... }

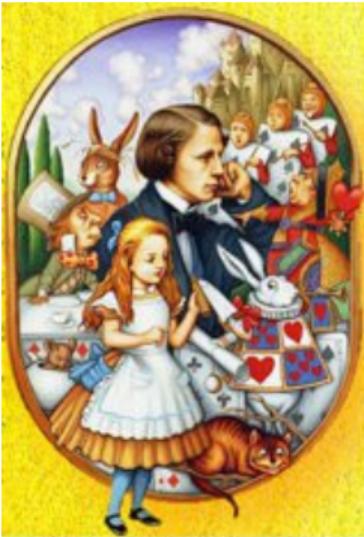
Grok Learning

- Grok Learning is the web-based programming environment we will be using for the duration of this subject
- All you need to access the system is a browser, an internet connection and your unimelb account
- Different modes of working in Grok:
 - code, run, mark
 - terminal

Lecture Outline

- 1 Who?
- 2 What?
- 3 Where and When?
- 4 How Do I Get Started?
- 5 Assessment
- 6 Getting Help
- 7 Tips for Winning at COMP10001

Where and When



"And how many hours a day
did you do lessons?" said Alice,
in a hurry to change the subject.

"Ten hours the first day," said the
Mock Turtle: "nine the next, and so on."

"What a curious plan!" exclaimed Alice.
"That's the reason they're called lessons,"
the Gryphon remarked:
"because they lessen from day to day."

Lewis Carroll

Where and When

- Lectures (three per week):

1	Tue	10.00am–11.00am
2	Wed	10.00am–11.00am
3	Thu	10.00am–11.00am

- All lectures are recorded, attendance is not compulsory. Some materials will be pre-recorded, please watch them as well!
- Tutorials (aka Workshops) (3x one-hour sessions per week, via zoom/on-campus (check your timetable))
the best place to ask questions
... starting **THIS WEEK!**

Workshop / Tutorial and Drop-In Zoom Sessions

- You should be registered for a particular tutorial classes and attend it Tue/Wed/Thu each week (1 hour duration each)
- Drop-in sessions are optional sessions run Tue/Wed/Thu 11am-12pm on Zoom (check “Zoom” on the LMS). There you can get help with any questions you might have either with lectures, tutorials or grok worksheets. You can attend any session that you want and you can attend more than one if you would like some extra help. We also provide Grok Tutor messaging, 11am-1pm Tue/Wed/Thu (may add more after Week 2).

Lecture Outline

- 1 Who?
- 2 What?
- 3 Where and When?
- 4 How Do I Get Started?
- 5 Assessment
- 6 Getting Help
- 7 Tips for Winning at COMP10001

How do I Get Started?

- Check out the LMS:
`canvas.lms.unimelb.edu.au`
- Log in to Grok Learning via the LMS “Assignments” tab
- Lecture slides, lecture recordings and code snippets from lectures will be made available on the LMS
- Take a look over the schedule for the subject from the “Subject Map” box
- Lecture recordings are accessible via “Lecture Capture” tab

Lecture Outline

- 1 Who?
- 2 What?
- 3 Where and When?
- 4 How Do I Get Started?
- 5 Assessment
- 6 Getting Help
- 7 Tips for Winning at COMP10001

Assessment I

- Your subject mark will be made up of:
 - Interactive Grok Learning worksheets: 10%
 - Projects ($\times 2$): 30%
 - Mid-term test: 10%
 - Final exam: 50%
- There will be rolling deadlines for Grok Learning worksheets, as listed on the LMS

Assessment II

- There are two “hurdles” for the subject: you must achieve at least 50% for the projects/interactive worksheets (at least 20 out of 40 possible marks)
AND at least 50% for the mid-term test/final exam (at least 30 out of 60 possible marks)

If you fail **either** component, you will fail the overall subject

Of course, you must also achieve at least 50% overall
(at least 50 out of 100 possible marks)

Lecture Outline

- 1 Who?
- 2 What?
- 3 Where and When?
- 4 How Do I Get Started?
- 5 Assessment
- 6 Getting Help
- 7 Tips for Winning at COMP10001

How do I Get Help?

- Make use of help within Grok (tutor messaging)
- Post a question to the Grok forums
- Talk to your tutor during your tutorial
- Attend drop-in Zoom consultations
- Email the lecturer

ekaterina.vylomova@unimelb.edu.au

How do I Get Help?

- Make use of help within Grok (tutor messaging)
- Post a question to the Grok forums
- Talk to your tutor during your tutorial
- Attend drop-in Zoom consultations
- Email the lecturer

ekaterina.vylomova@unimelb.edu.au

If you are struggling, don't be shy about asking for help; similarly if you are experiencing documentable hardship and unable to meet submission deadlines, let us know **at the time**

Collaborate/Discuss!

- Discuss problems with others, develop subject (domain-specific) terminology
- Explain to others (you will learn it much better and see knowledge gaps)
- Write code, practice more!

More References

- Check “**Student Support Services**”
- Check the “**Getting Help**” LMS page
- Check **Python Resources**

Lecture Outline

- 1 Who?
- 2 What?
- 3 Where and When?
- 4 How Do I Get Started?
- 5 Assessment
- 6 Getting Help
- 7 Tips for Winning at COMP10001

Tips for winning at COMP10001

- Balanced workload per week, e.g.:
 - Tutorial (3 hours attendance)
 - Lectures (3 hours attendance + pre-recorded materials)
 - Study (2 hours review/reading/study group)
 - Form an informal study group and copiously share ideas, and **non-embargoed** code
- Attend, listen, ask, and share, but above all do, do, do!

Tips for winning at COMP10001 I

- All lecture content and “non-optional” Grok worksheets are examinable
- Advanced lectures and “optional” worksheets are **not** examinable
- Access Grok Learning using your unimelb credentials via the Grok Worksheets on the Assignments page of the LMS

Tips for winning at COMP10001 II

- All “tutesheets” from workshops will be made available (complete with a solution) at the end of the corresponding day, on completion of all workshops
- Within Grok Learning, you will be able to access the COMP10001 materials, and also see materials for other courses offered by Grok Learning. Note that you might have free access to some other Grok Learning courses, but none of it is examinable or required for this subject.

Tips for winning at COMP10001 I

- All lecture slides will be made available on the LMS ahead of time (but may be tweaked slightly leading up to/after the lecture)
- Lectures and workshops start 5 mins later and end 5 mins earlier than advertised
- All lectures will be recorded (audio and screen scrape) ... but try to come along to ask questions and get the full lecture experience
- Expect things to move faster and marks to be harder to get than in high school

Tips for winning at COMP10001 II

- When learning programming, constant “practice” is the key to success
- Never share any **examinable** code with your fellow students (not on the forums, not via email, not via shared machines, ...)
- If you need help, avail *yourself* of the various sources of assistance; don’t expect us to come chasing you

Academic Honesty (Very Important! Please Watch it!)



*Please
Watch
Pre-recorded
Materials
Here!*

Things to do before the Next Lecture

- Check that you can access the **subject LMS site**
- Check that you can access **Grok** (using Grok worksheets on Assignments page of the LMS)
- Post to the Grok forum (personal testimonial, computing-related material, but not code for assessment...)
- Attend your first workshop today:-)