



THE UNIVERSITY OF
WESTERN
AUSTRALIA

CITS1401

Computational Thinking with Python



THE UNIVERSITY OF
WESTERN
AUSTRALIA

Lecture 0

Introduction

My introduction

- Unit Coordinator: Dr. Ghulam **Mubashar** Hassan
- Consultation time : 2 – 3 pm Tuesdays
- My research areas: Artificial Intelligence, Machine Learning interdisciplinary problems & Engineering Education
- Website: www.csse.uwa.edu.au/~00080148/
- Office: **CSSE Room: 2.12** and **MS Teams**

All of you are part of MS Teams. Please use your Pheme details to access it.

Teaching team

- Teaching team
 - *Associate Lecturer*
 - Dr. Zulqarnain Gilani
 - *Lab facilitators*
 - Sania Zahan
 - Mustafa Saeed
 - Nick Hodgskin
 - Jasper Paterson
- Admin/enrolments/labs/etc.
 - *CSSE Reception (Hass Barzani) or admin-csse@uwa.edu.au*

What is CITS1401 About?

“Computational thinking is the thought processes used to formulate a problem and express its solution or solutions in terms a computer can apply effectively”

Cansu, S. K., & Cansu, F. K., 2019

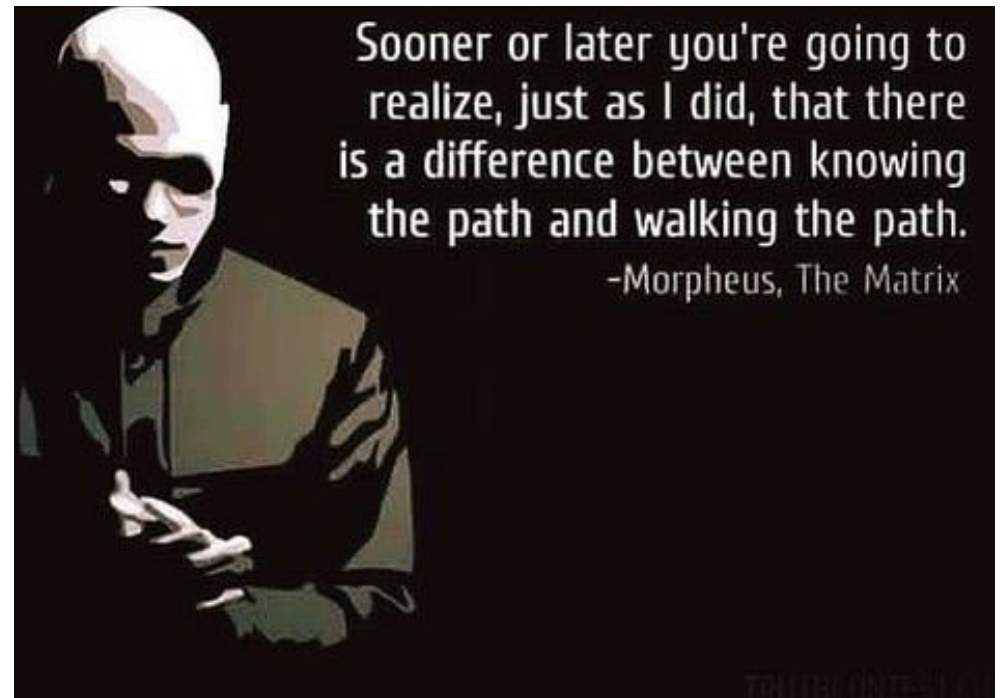
- CITS1401 is about computer-based problem solving
 - *How to formulate the problem in a computer language as series of steps*
- Will say a little about computers and how they work, and also about how to solve problems using programs
- Shall be using Python 3 as our computer language
 - *Please do not use Python 2. Related dialect, but incompatible*

Course Outcomes

- Developing computational thinking skills:
 - *Decompose*: how to divide large problem into small parts and solve them “*divide and conquer*”.
 - *Pattern recognition*: recognizing common tactics to solve set of problems.
 - *Abstraction*: generalizing the problem by reducing avoidable details.
 - *Algorithm*: how to formulate ordered step-wise approach to solve the problem.
- Developing programming skills:
 - *Be able to write a program in Python 3 to:*
 - Solve small problems
 - Automate repetitive computational tasks

Both skills are transferrable

Teaching Strategy



Why Python ?



- Most popular language
<https://www.northeastern.edu/graduate/blog/most-popular-programming-languages/>
 - *Easy to learn*
 - *Large library*
 - *Extensively used in:*
 - Startups
 - Artificial intelligence
 - Data science
 - Financial services
 - Interdisciplinary fields

Resources

- “Python Programming: An Introduction to Computer Science”, 3rd Edition, *John Zelle, Franklin Beedle*.
- “Starting out with Python” 4th Edition, *Tony Gaddis*
- All CITS1401 resources (including PDFs of the lectures) can be found on the LMS page for the unit.
 - *You need to be enrolled in the unit to see the page.*
- All assessments’ submissions will be made on [Moodle](https://www.jinhong.org/moodle/login/index.php) (<https://www.jinhong.org/moodle/login/index.php>) which is a similar platform as LMS but can run and test your code. More details will be shared via an announcement on LMS.
- Students need to follow course page on [LMS](#) as well as [Moodle](#).

Organisation

- 2 x 1hr lectures a week and 1 x 1hr Workshop
 - *Both lectures and workshop slots will be treated in similar manner. The contents of workshop are embedded in the lectures to make them more interactive.*
 - *The recorded lectures/workshops will be made available on LCS.*
 - *The lectures slides you find on LMS do not necessarily correspond to timetabled lecture/workshop slots.*

Organisation

- 1 programming lab per week (2 hrs)
 - *Lab facilitator is available to assist*
 - *There are 12 face-to-face and 3 online lab sessions*
 - *Starts Week 2*
- Check your Timetable for your lab session
 - *30 hours of multiple time slots across the week*
 - *feel free to drop in any lab session but registered students will be preferred*
 - *maximum number of students may be restricted due to COVID-19 restrictions in face-to-face sessions*

COVID Contingency Plan

Our priorities are:

- To ensure your safety and well-being on campus
- To support you to progress in your studies and ensure the best teaching experience possible

Keep up to date and be prepared:

- Please check the unit LMS site and your emails regularly (at least once a week) to ensure you do not miss an announcement
- If you have difficulty accessing suitable technology for online learning, please contact the Library for support at the beginning of the semester and advise unit coordinator

In case of a snap lockdown, or campus shutdown:

- We will move to online teaching

Classes

- Lectures/Workshops will be recorded and delivered via LCS accessible by LMS
- Labs will be shifted to online mode and will be delivered by MS Teams

Assessments

- **Note: Assessment items and weightings may need to be modified during semester as a result of the impact of COVID.**
- All assessments except final exam are already required to be submitted online.
- The arrangements of final exam will be informed.

COVID Contingency Plan

If you are unwell:

- Get in touch with the unit coordinator and Student Office as soon as possible
- Follow University and State Government protocols

If you have any other concerns or questions, please contact your unit coordinator or the EMS Student Office as early as possible (enquiries-ems@uwa.edu.au).

If you need further support with online learning, please use the resources at:

- <https://www.uwa.edu.au/students/Support-services/Learning-online>
- <https://www.uwa.edu.au/students/Support-services/Academic-support>

If you need support with IT issues, please contact the Library at:

- <https://www.uwa.edu.au/library/Help-and-support/IT-and-printing-support>

IT resources are also available at: <https://www.it.uwa.edu.au/it-help>

University COVID advice and updates are available at:

- <https://www.uwa.edu.au/covid-19-faq/Home>

Labs - Expectations

- Five labs are **assessed** and rest non-assessed. Lab 00 explains Moodle and must be attempted.
- If you want to do well in the unit you should complete labs regularly.
 - *Some learning in the unit, particular related to computational thinking skills, will only take via labs.*
- Students are encouraged to start working to solve the lab as soon as they are released.
- You are welcome to attend as many lab sessions as you want
 - *preference to those timetabled to be there*
- Feel free to use discussion forum on Moodle. Teaching team will regularly check the forum during working days. Estimated time to respond your queries is 2 working days.

Labs - Expectations

- You are required to install Thonny on your own computers. Thonny is already available in all lab computers of EMS. (You may bring your laptops and use Thonny while watching the lectures and workshops)
- **Lab time is your time to seek help from lab facilitators**
- *The contents covered in labs are part of the course and it may be more than you have covered in the lectures*

Programming Environments

- In the lab you will use Python 3.5 (or above) via the Thonny IDE
 - *An integrated software development environment where you can write, edit, execute and debug programs*
- Thonny is student oriented. It is a free software available for all major operating systems such as Windows, OS, Linux. Python 3.5 or above is built in
 - *Not phones or tablets (Android or iOS)*
- You can download Thonny from <http://www.thonny.org>

Assessment

- Assessment is based on both
 - *Understanding of fundamental concepts*
 - *Practical computational thinking and programming skills*
- Final Exam (worth 55%) *TBA by Exam Office*
- Two programming projects (worth 25%)
 - *Project 1 due **Tue. 9:00 am of Week 8** (worth 15%)*
 - *Project 2 due **Fri. 5:00 pm of Week 12** (worth 20%)*
- Labs (worth 10%):
 - *Five lab quizzes (worth 2% each) due **Fri. 5:00 pm** in two weeks after their release*

Getting Help

- Discussion Forum on Moodle
- Weekly Consultation hour
- 30 hours of Labs
- Textbook

- Above all, seek help early



Svengraph, Wikimedia

Do Something Useful in Week 1

- Get your pheme login and password
- Get an access to Moodle's server as soon as details are sent to you by email.
- Organize your UWA email account
- Obtain your timetable (online)
- Get familiar with the CITS1401 LMS and Moodle websites
- Install Thonny (it comes with recent version of Python)

Other Stuff

- You may read other textbooks or lectures to improve your understanding of fundamental concepts or learn more
- I have set slides in Century Schoolbook font (with some Courier and Arial for computer code and meta-language). If you have trouble reading it, please let me know
 - *Accessibility is important*

Other Stuff

- “10 Signs You Will Suck at Programming”
 - *Article made available on [LMS->Interesting Things](#)*
 - *Has really great advice about what you need to succeed at programming.*
 - *READ IT*
- Engage with the unit!!!
 - *From the last years, I observed that if you watch lectures and generally engage with the unit regularly, you will do better.*



PhoebeA - Redbubble

Acknowledgements

- It is important to acknowledge the PPT slides for this unit are based on a slide deck supplied by *John Zelle* (textbook author), though modified, augmented and reordered by *Ghulam Mubashar Hassan* and *Michael Wise*.