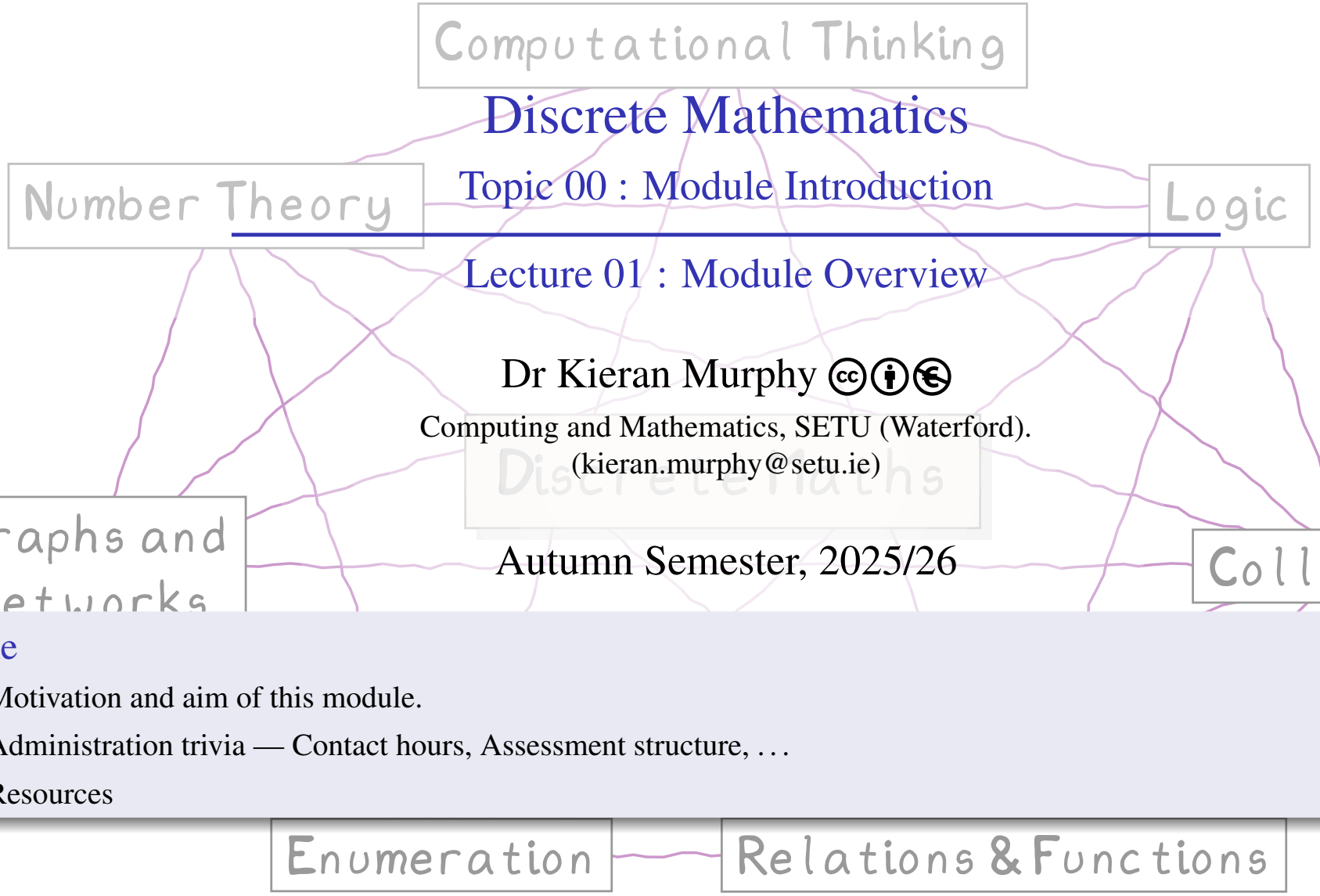


Discrete Mathematics

Topic 00 : Module Introduction

Lecture 01 : Module Overview

Dr Kieran Murphy
Computing and Mathematics, SETU (Waterford).
(kieran.murphy@setu.ie)

Autumn Semester, 2025/26

Outline

- Motivation and aim of this module.
- Administration trivia — Contact hours, Assessment structure, . . .
- Resources

Outline

What? (Aim)

Aim, as per the Module Descriptor . . .

This module provides a solid foundation of selected topics in discrete mathematics related to computing and information sciences. The topics are covered in an elementary manner in order to reinforce understanding of concepts and improving algebraic problem-solving skills so that the student can effectively proceed with their study of a degree programme in computing.

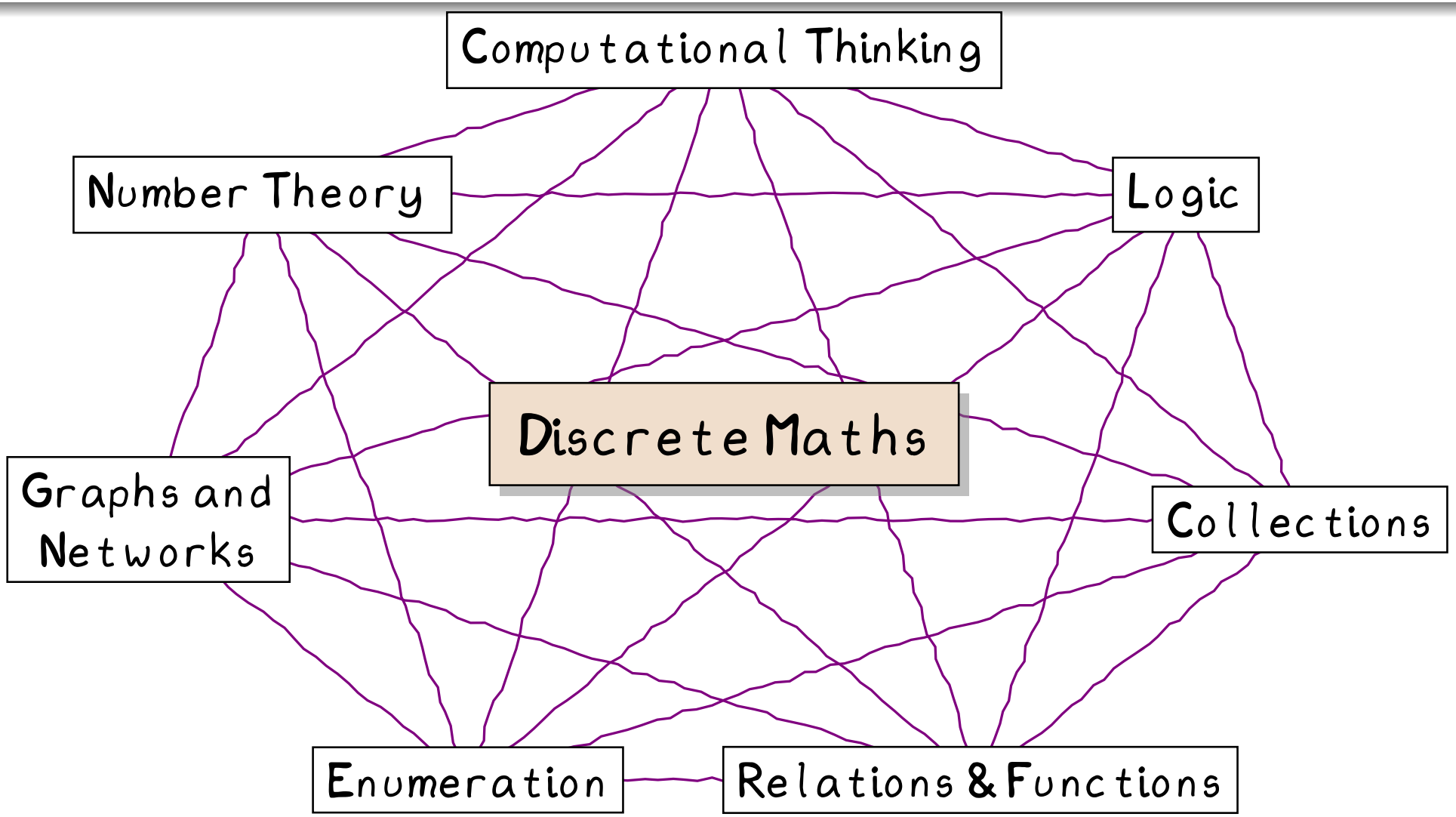
Translation (Informal Aims)

1. Reason logically — aim for precision and correctness over speed.
2. Develop and manipulate theoretical models

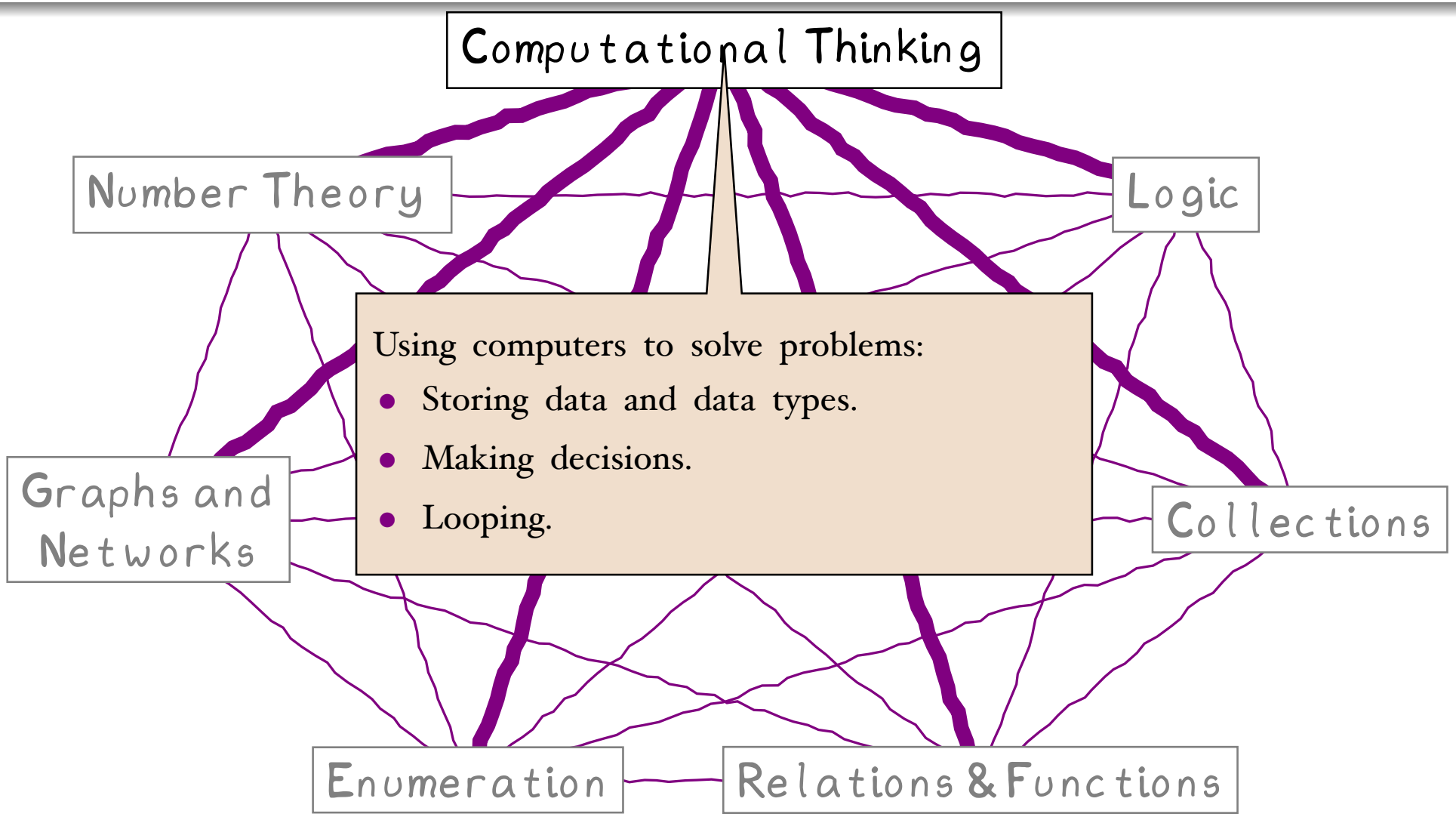
   — a set is a collection of things, a relation is a collection of pairs of things, a graph is a collection of things with pairwise connections, etc..
3. Translate

   computing concepts/implementation (Python) $\leftrightarrow$ theoretical models (mathematics).

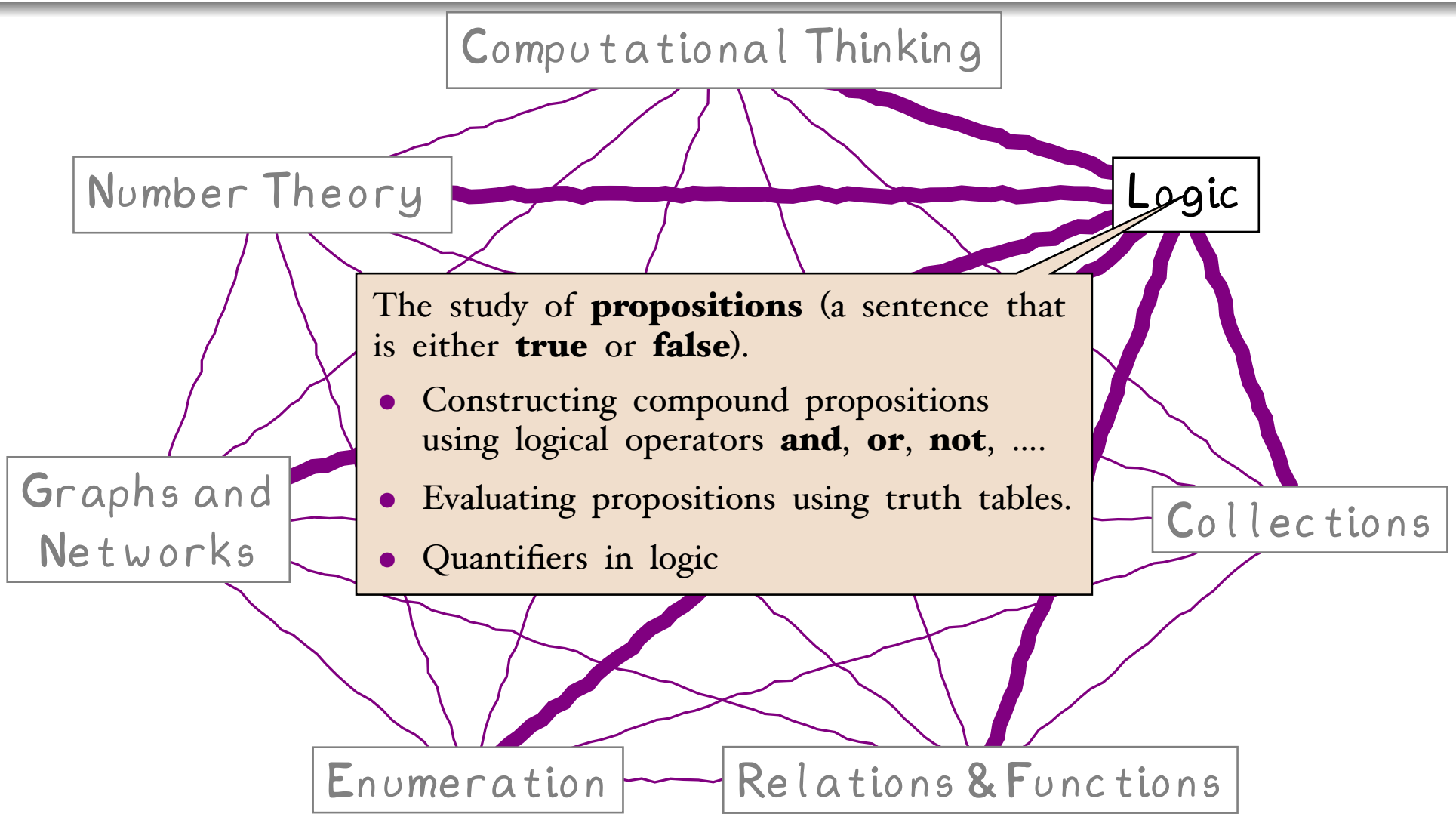
What? (Discrete Mathematics)



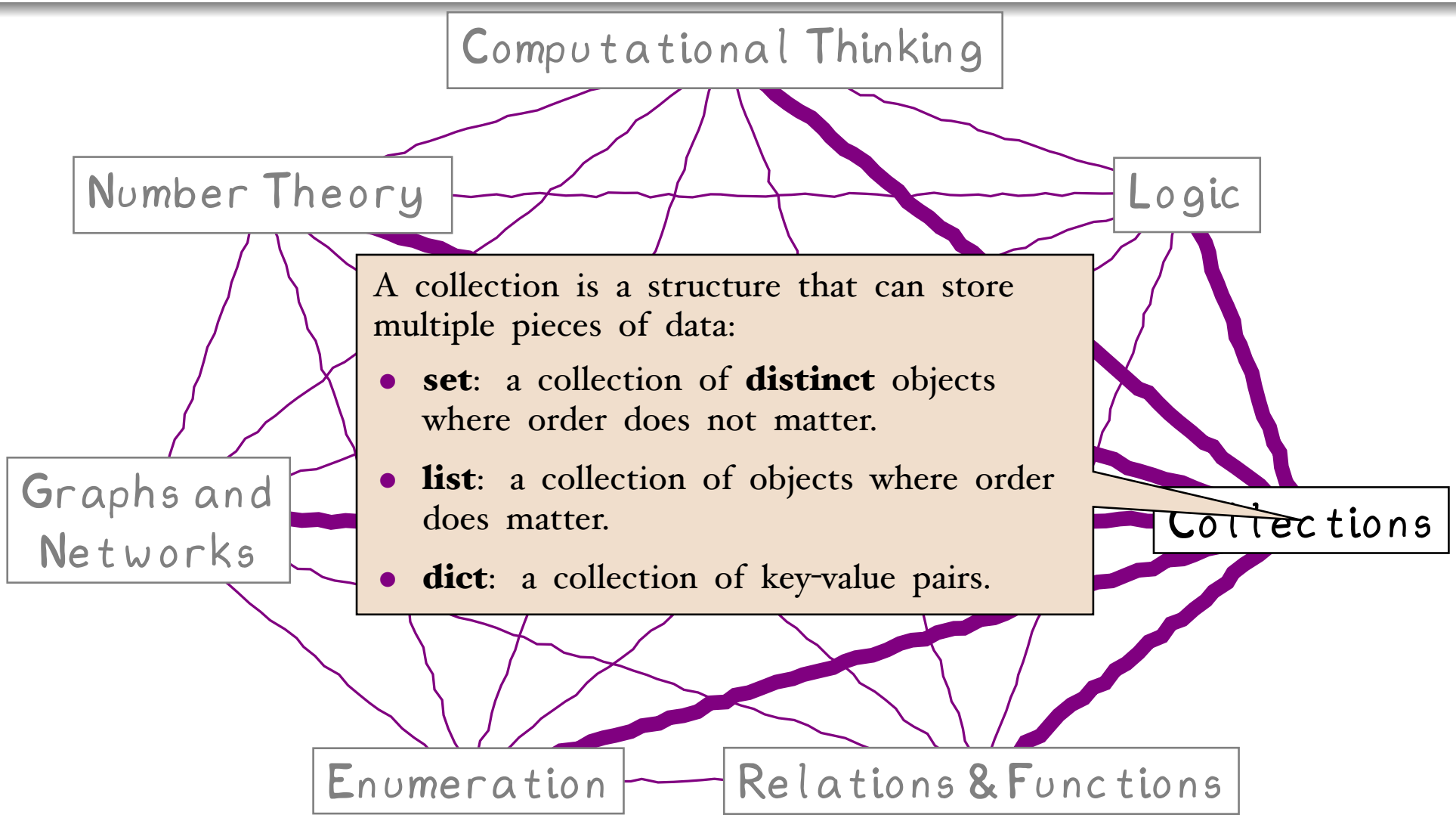
What?


Computational Thinking
Number Theory
Logic
Using computers to solve problems:
Storing data and data types.
Making decisions.
Looping.
Graphs and Networks
Collections
Enumeration
Relations & Functions

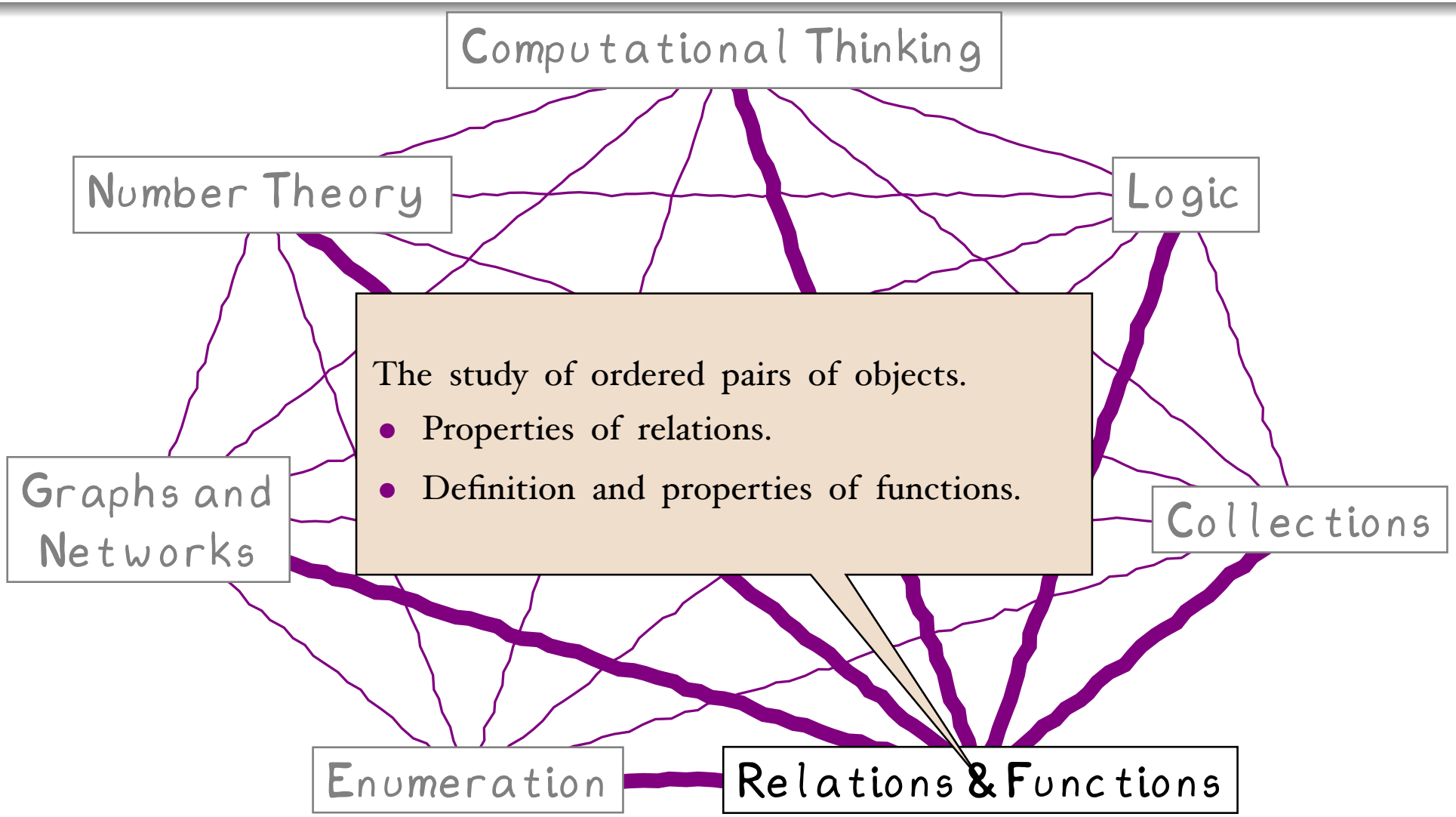
What?


Computational Thinking
Number Theory
Logic
Graphs and Networks
Collections
Enumeration
Relations & Functions
The study of **propositions** (a sentence that is either **true** or **false**).
• Constructing compound propositions using logical operators **and**, **or**, **not**, ...
• Evaluating propositions using truth tables.
• Quantifiers in logic

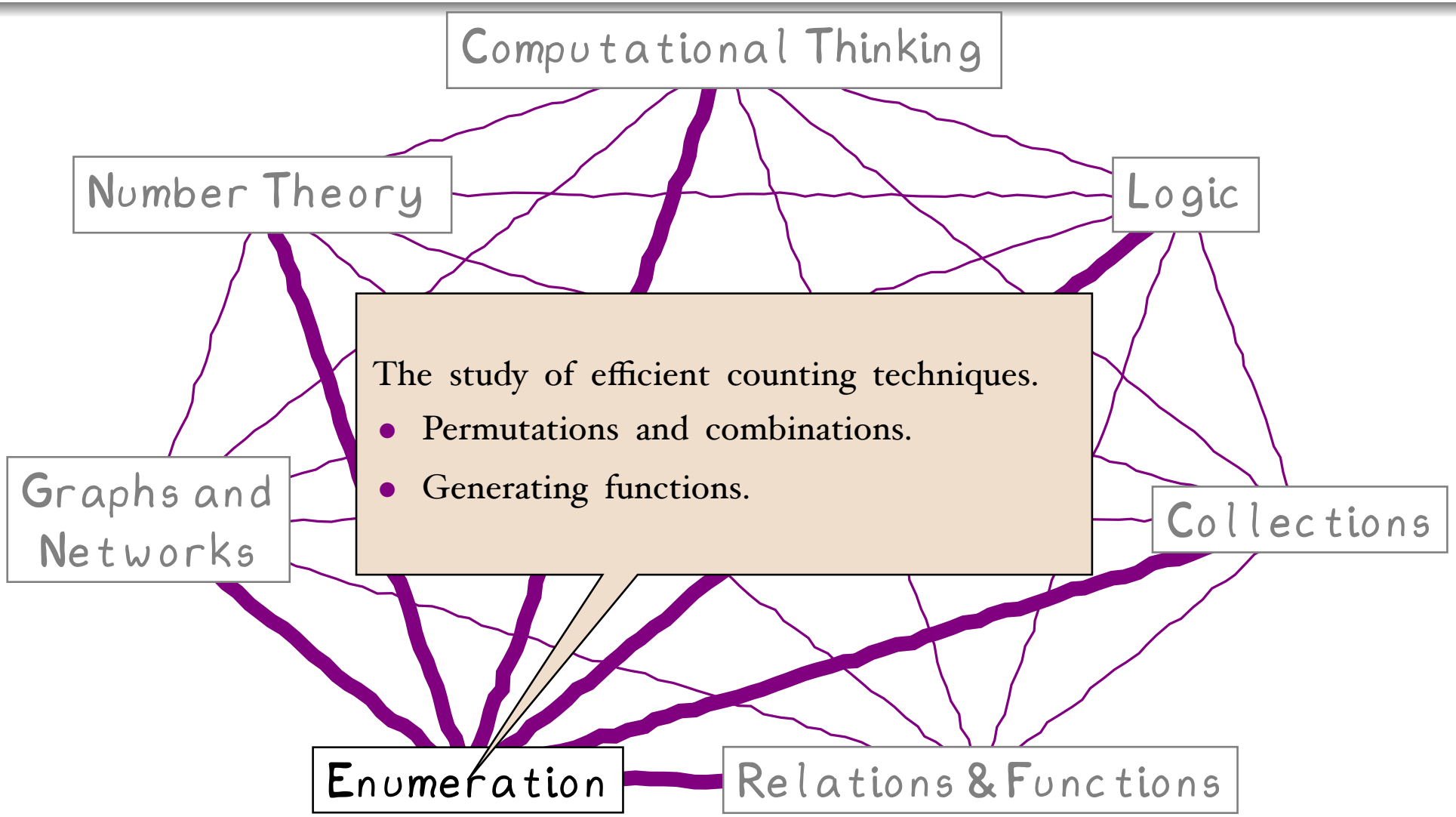
What?


Computational Thinking
Number Theory
Logic
A collection is a structure that can store multiple pieces of data:
• set: a collection of distinct objects where order does not matter.
• list: a collection of objects where order does matter.
• dict: a collection of key-value pairs.
Graphs and Networks
Collections
Enumeration
Relations & Functions

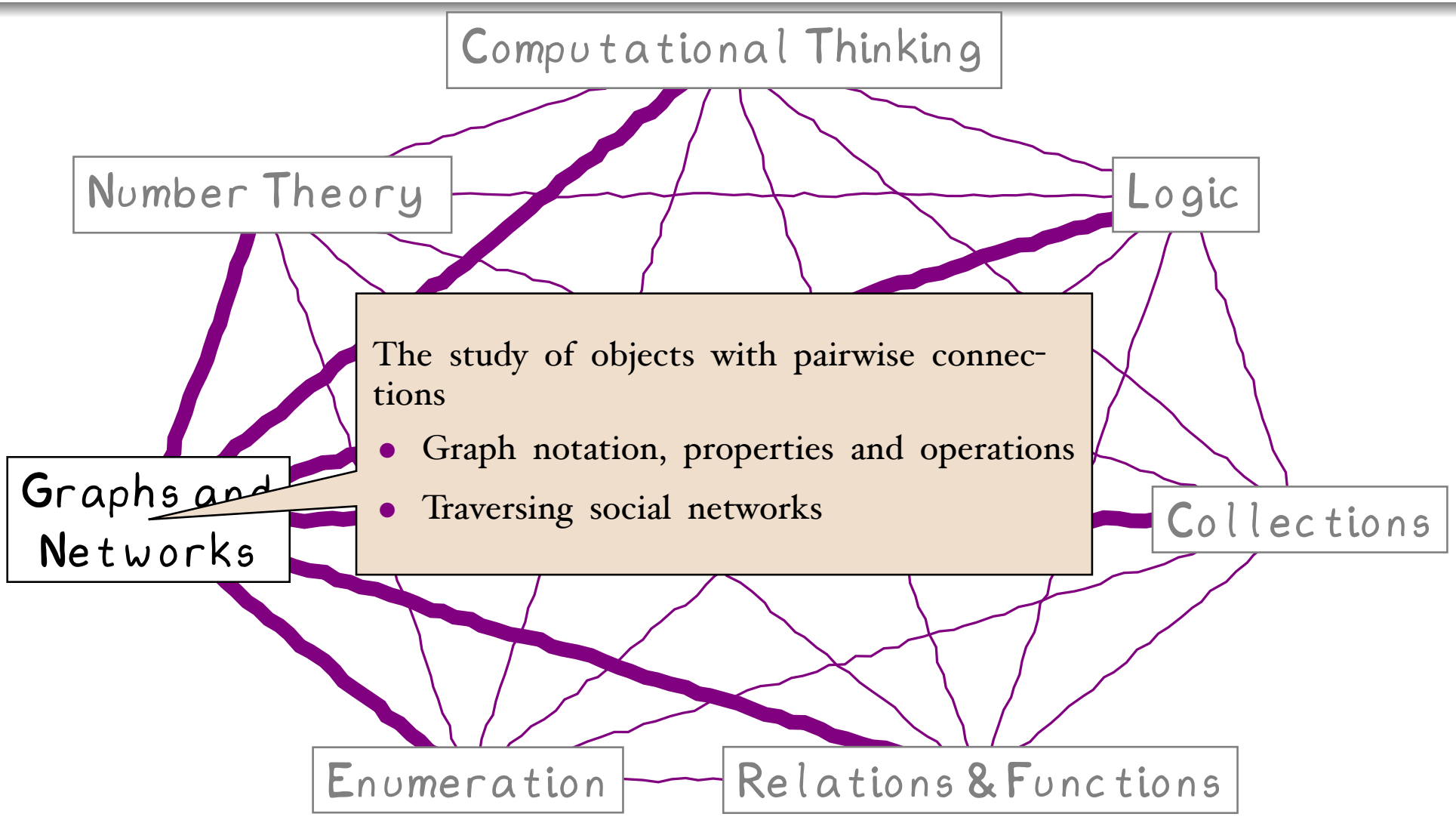
What?


Computational Thinking
Number Theory
Logic
The study of ordered pairs of objects.
• Properties of relations.
• Definition and properties of functions.
Graphs and Networks
Collections
Enumeration
Relations & Functions

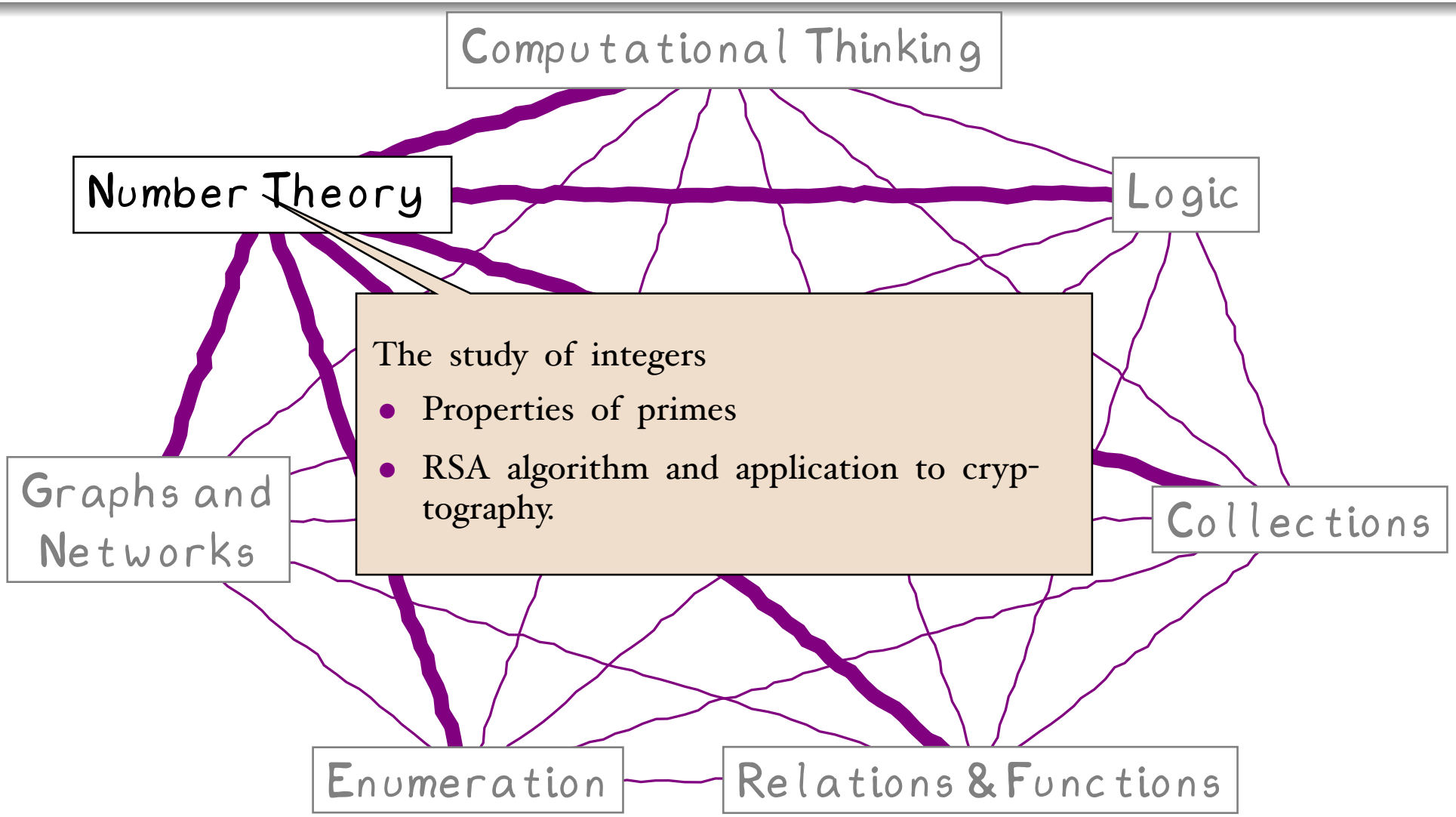
What?


Computational Thinking
Number Theory
Logic
The study of efficient counting techniques.
• Permutations and combinations.
• Generating functions.
Graphs and Networks
Collections
Enumeration
Relations & Functions

What?


Computational Thinking
Number Theory
Logic
The study of objects with pairwise connections
Graph notation, properties and operations
Traversing social networks
Graphs and Networks
Collections
Enumeration
Relations & Functions

What?

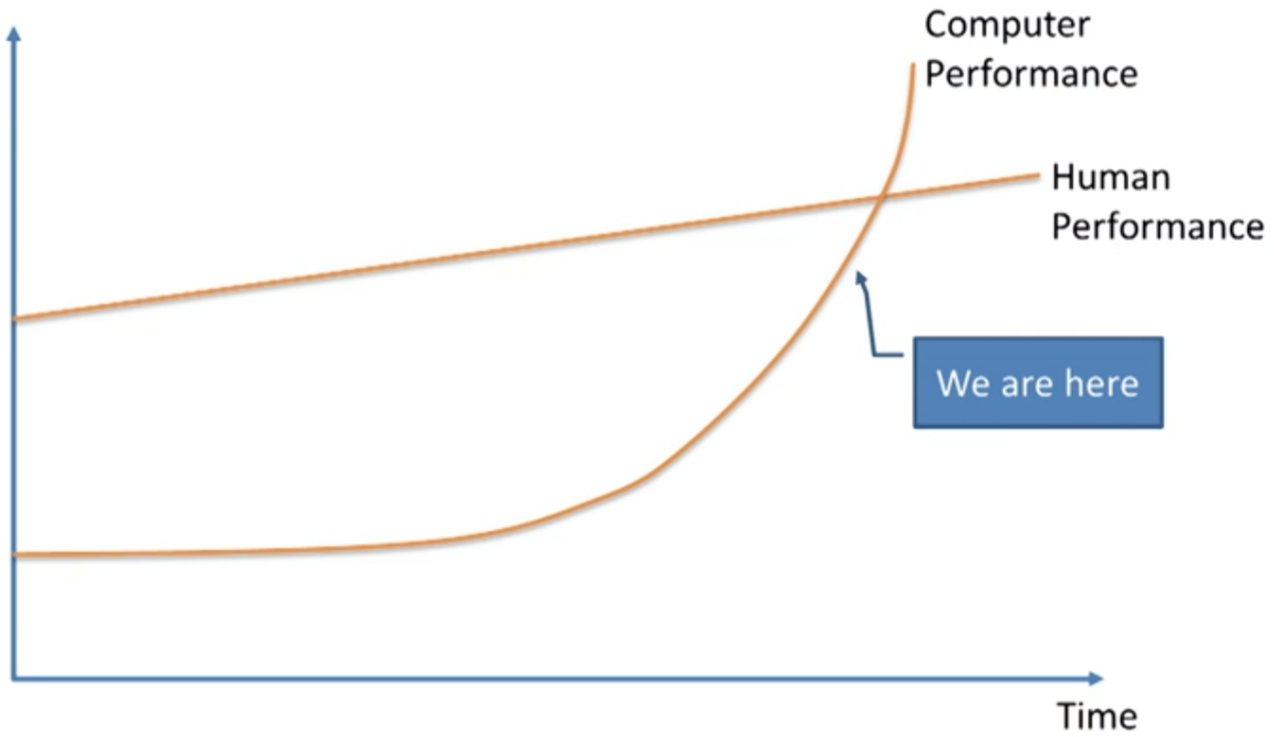

Computational Thinking
Number Theory
Logic
The study of integers
Properties of primes
RSA algorithm and application to cryptography.
Graphs and Networks
Collections
Enumeration
Relations & Functions

Why?

Many, many reasons . . . pickinig one* . . .

Machine learning is the future of computing

Discrete mathematics is the core of machine learning.



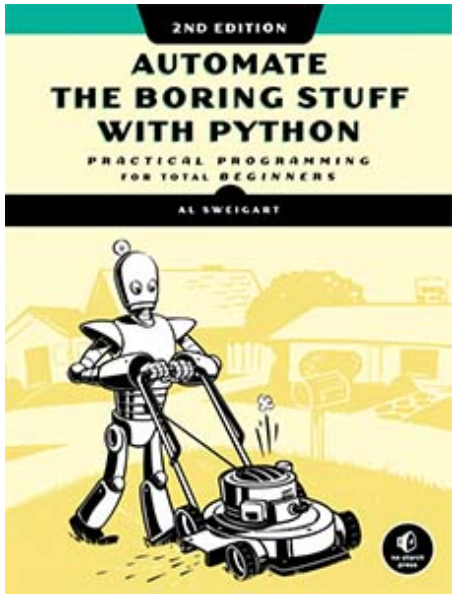
*The AI Revolution: Our Immortality or Extinction
waitbutwhy.com/2015/01/artificial-intelligence-revolution-2.html

Why? (Aim)

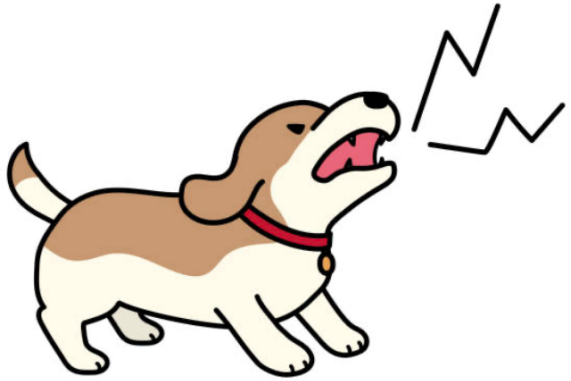
Computers are faster and more accurate* than us, they don't get board/distracted†

$\Longrightarrow$ Get computers to do the hard (and/or boring) stuff

See Al Sweigart's books



Or our view on life



Why have a dog and bark?

Exception to this are the generative text models (chatGPT, Bard, . . .) where accuracy is replaced by "most likely/probable".

†Well, if you don't count Teslas, in FSD mode, driving over pedestrians . . .

How? (Contact Hours)

- Three lectures per week
  - Cover concepts, definitions, examples, etc.
  - BUT feel free to stop me and ask questions at any point.
  - **You should print out notes in advance of lecture, or have access to them during class.**
  - Ideally you skimmed over the notes in advance of lecture.
  - **Take notes during lectures.**
- One tutorial per week
  - Review of exercises based on the material covered in the lecturers.
  - **You need to have printout of tutorial sheets in advance of lecture.**
  - Ideally you have attempted/completed some/all questions in advance of tutorial and you are just attending the tutorials to show off.
  - Online quiz for self review at end of each topic.
- One practical per week
  - Using Python (via *colab notebooks*) to demonstrate implementation details of discrete mathematics concepts.
  - Introduce programming in Python — never have too much programming.
  - **You need to upload notebook by end of week (Saturday 11:00pm).**

How? (Assessment Structure)

75% End of Semester Exam

Current plan (this is subject to change so ask about this in week 10!)

- 4 questions (typically 3–5 parts per question. Answer all questions (i.e. no choice).
- Tend not to have question per topic.
- 80% same material as last year — see pre-Covid exam papers, but there may still be some differences in format/style of questions as will the relative emphasis/weighting of the different topics.

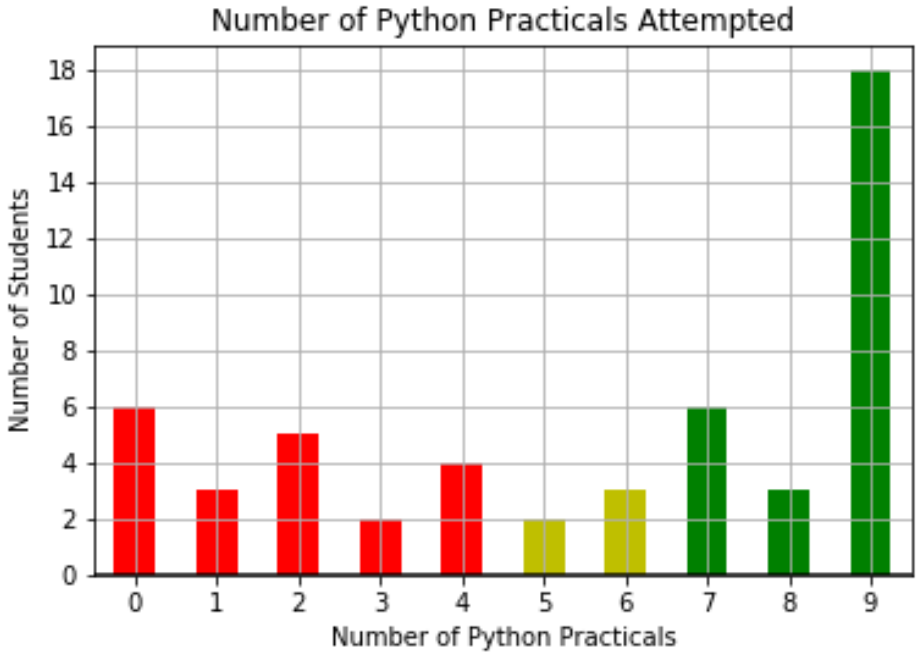
25% Continuous Assessment

- Practical work based on 10 python practicals and 2 online class tests[‡].
- In theory[§], weekly assignments, are graded in advance of next week.

[§] In practice, I ~~may~~ will fall behind a bit.

A Brief Look at 2019/20 Results

- 52 students enrolled, but only 28 passed! $\Longrightarrow$ pass rate of 53.8%.
- Of the 32 students who attempted at least 5 practicals, 25 passed $\Longrightarrow$ pass rate of 78.1%.
- Of the 27 students who attempted at least 7 practicals, 24 passed $\Longrightarrow$ pass rate of 88.8%.



Keep up with the material:

- Read notes before & after lecturers
- Attend practicals and upload practical work on time.
- Attempt tutorial questions.

Note "Attempted" $\neq$ "Completed correctly"

A Brief Look at Last Year's Results (2024/25)

- 88 students, pass/compensation rate of 64%. (better but can still improve)
- Comparison of Practical Work (CA) vs Final Exam:
  - One student passed while failing the CA

    It is possible to pass this module without doing the practical work but the odds are against you.
  - Average grade on CA was 66.1% while on Final Exam was 47.5%

    The CA is graded easier (a carrot to help keep you motivated during the semester).
  - Average number of practicals attempted last year was 6.6.

    While (like in 2018/19) the more practicals a student attempted the more likely they passed the module, but if you miss a week or two it is not "end of the world".

Who?

Dr Denis Flynn

Room 327
051–30 2068
Denis.Flynn@setu.ie

Background

- PhD in Applied Mathematics and Civil Engineering.
- BSc (H) Physics.

Academic Interests

- Dynamical systems, in particular systems with hysteresis
- Game development
- Langauges: C/C++, Python, Java

Dr Kieran Murphy

Room 313
051–30 2055
Kieran.Murphy@setu.ie

Background

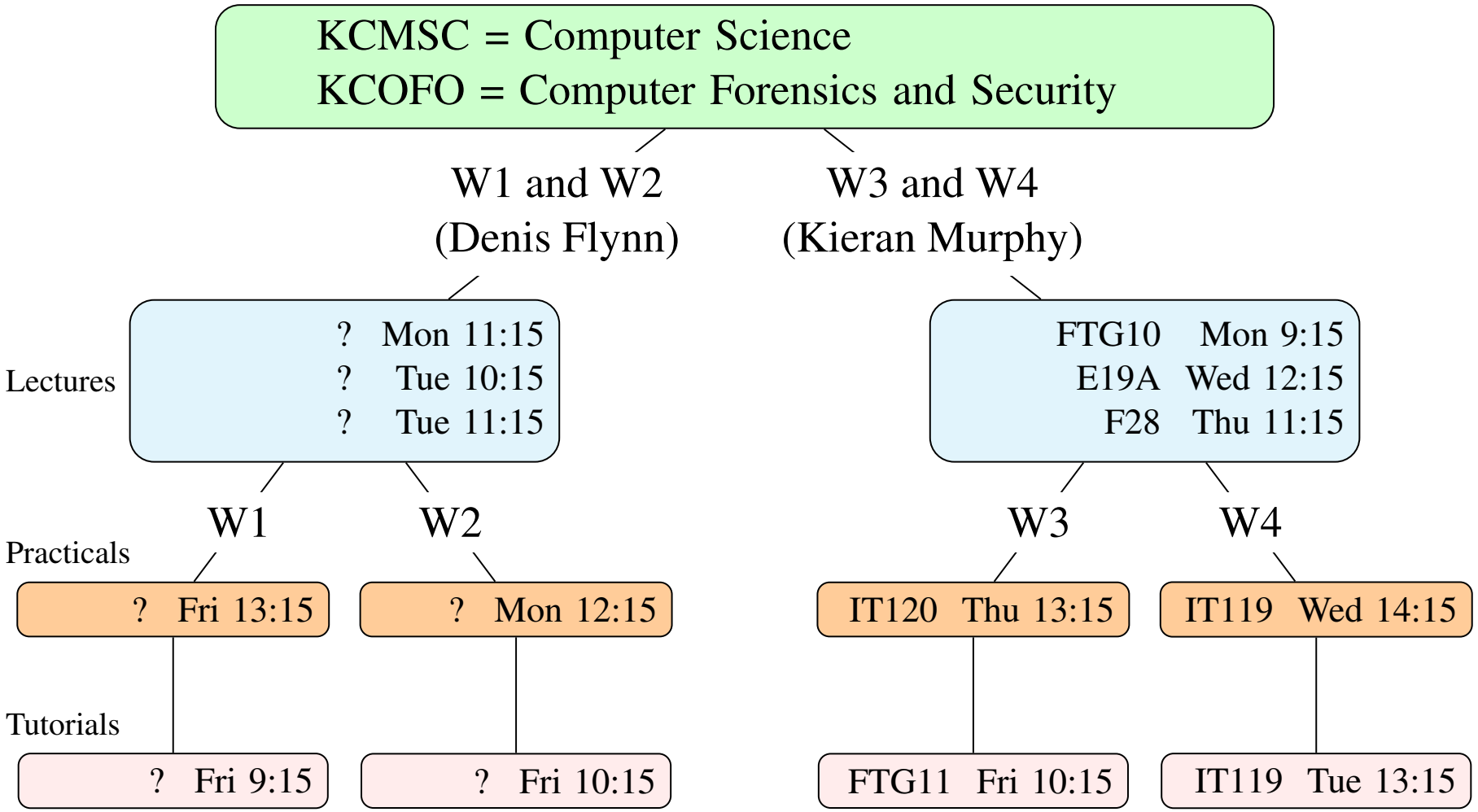
- PhD in Applied Mathematical Sciences
- BSc (H) Applied Mathematics.

Academic Interests

- Dynamical systems, in particular numerical analysis
- Game development
- Langauges: C/C++, Python, Java

When?

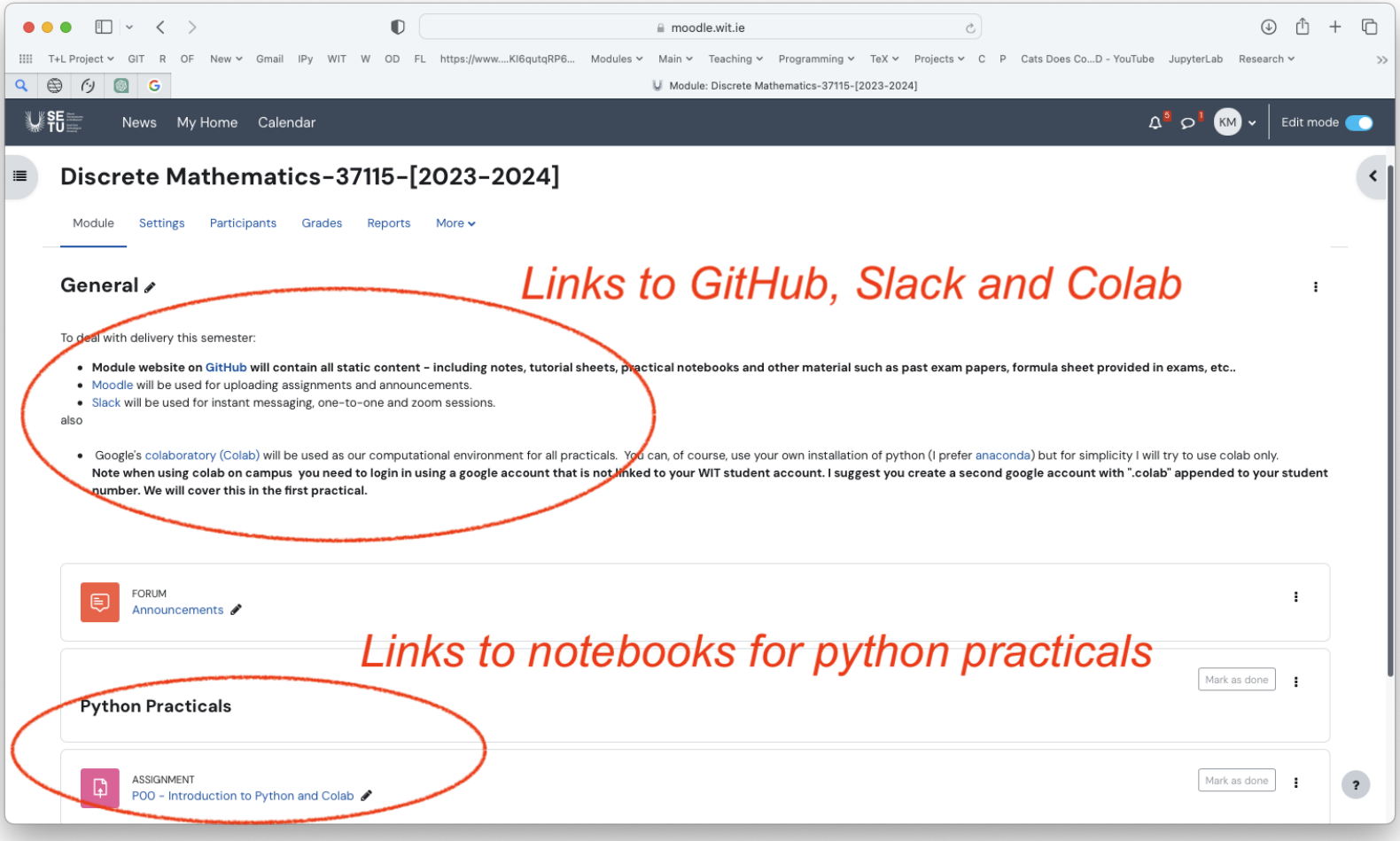
(Correct as of 11 Sep 2025)



Outline

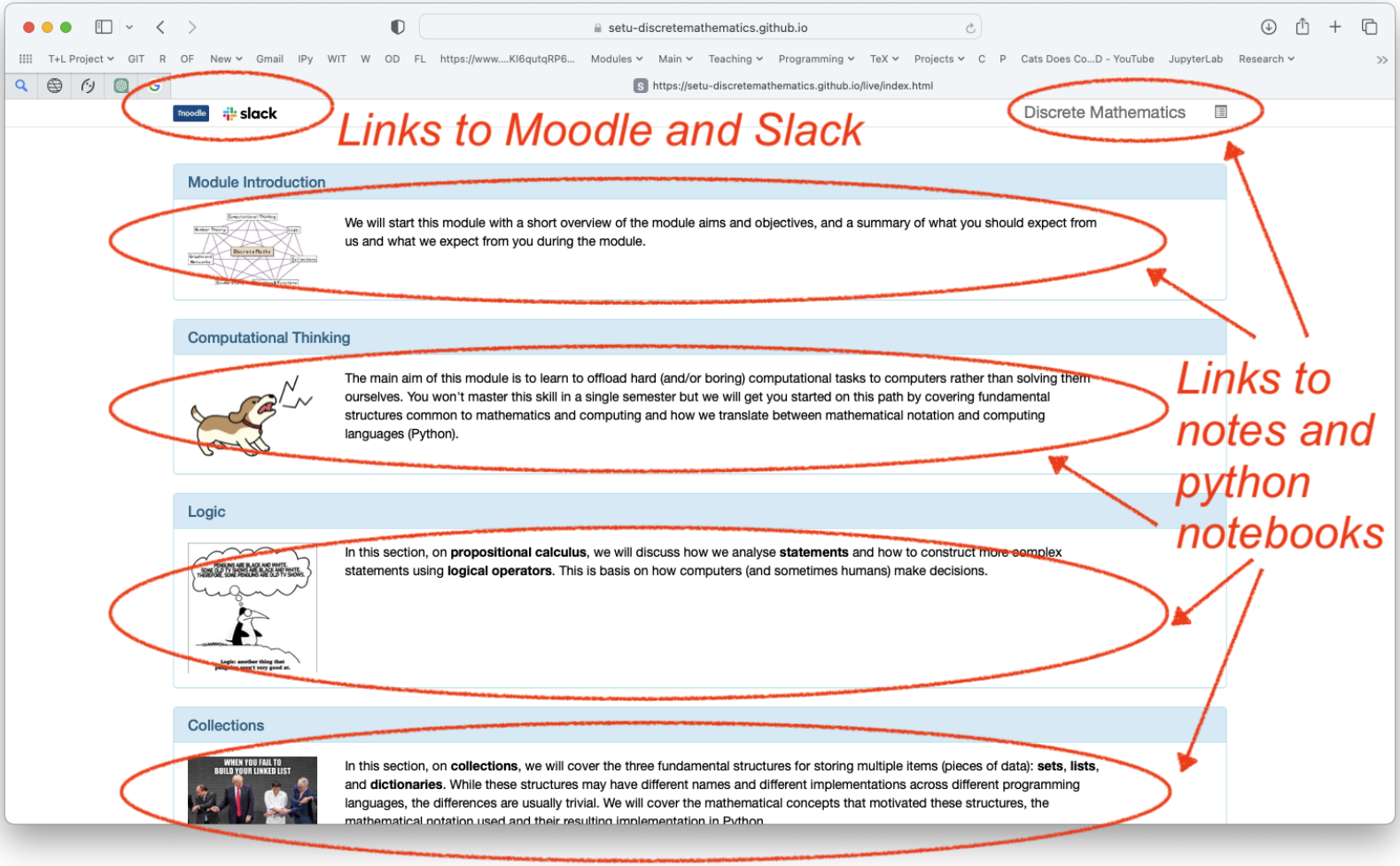
Resources — Moodle

- URL: moodle.wit.ie/course/view.php?id=217956
- Used for all notices, assignment and practical work submissions.



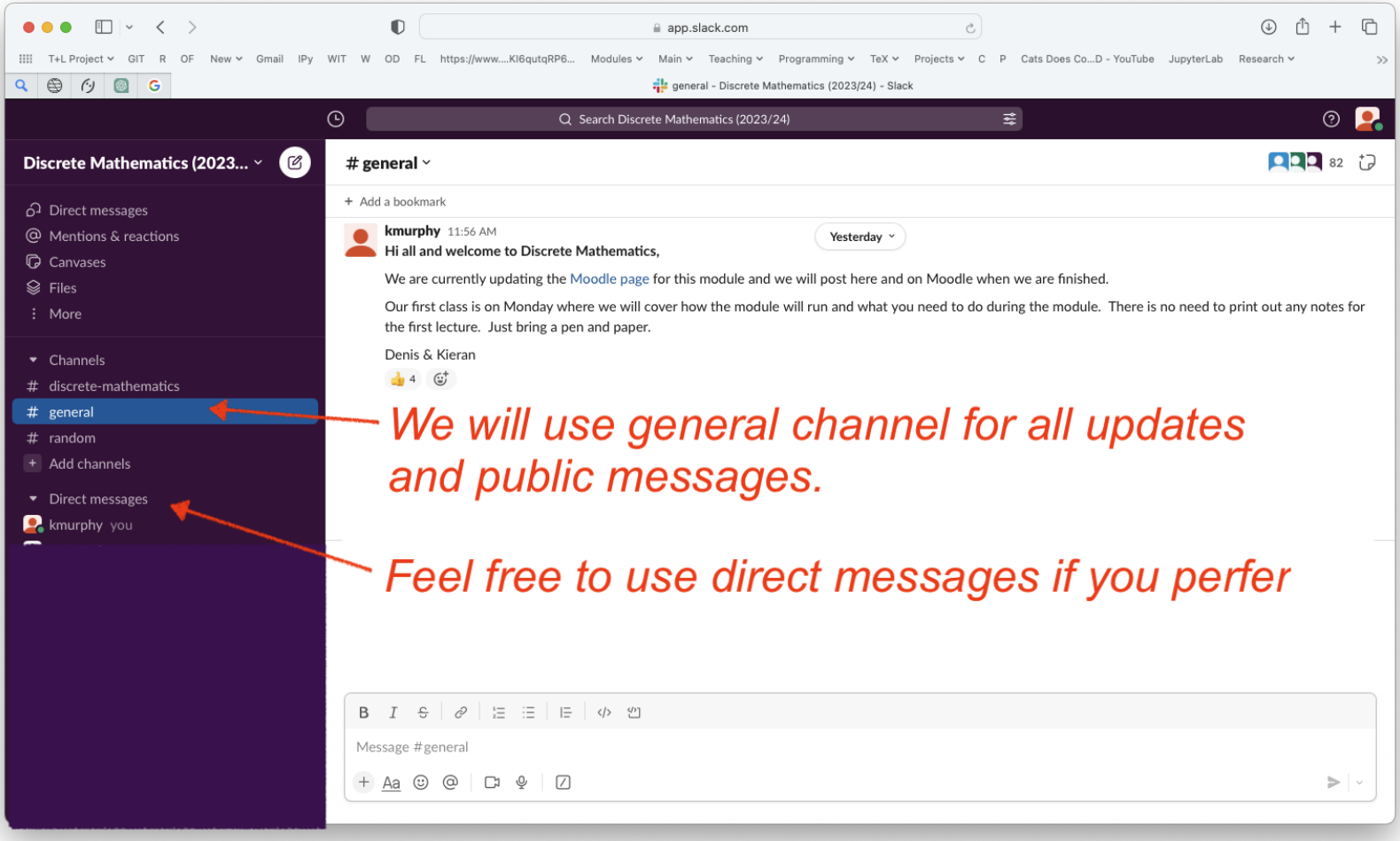
Resources — Github

- URL: SETU-DiscreteMathematics.github.io/live
- Used for all content (slides, notebooks, tutorial sheets).



Resources — Slack

- URL: discretemathe-7co3349.slack.com
- Used for instant messaging, one-on-one sessions, etc.



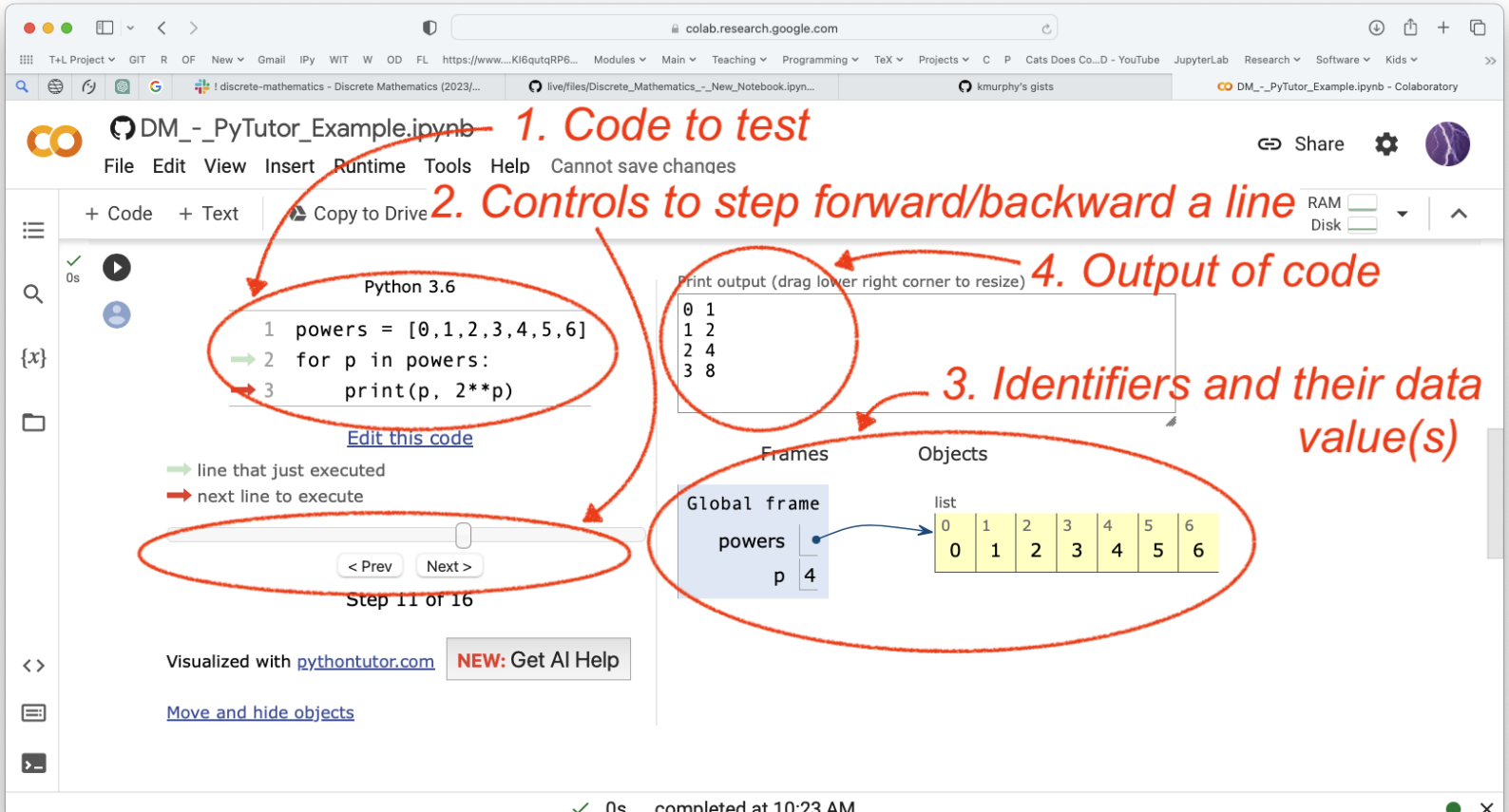
Resources — Colab

- We will use the online Google Colab[¶] environment to code in python for all of our practical work.
- You can open a notebook from these slides by clicking the "OPEN in COLAB" icon or clicking/scanning the QR code

[¶]Alternatively, if you want to install python on your laptop you could use the anaconda distribution from www.anaconda.com (just install the latest 64-bit, version 3.+).

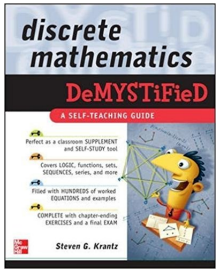
Resources — PyTutor

- PyTutor (pythontutor.com) is a website that helps programmers to learn Python, Javascript, C/C++, and Java by visualising code execution (shows what happens to data as code runs, line by line).
- We will run PyTutor from within Colab so will demo PyTutor when we cover Colab (or click/scan the QR code).



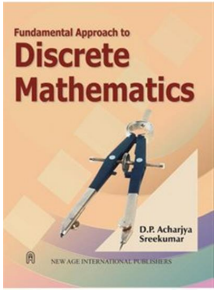
Text Books

I like the following textbooks on discrete mathematics and expect that my notes will overlap significantly with these books. I do encourage you to read* them†, however, be aware they may use different notation or cover different topics.



Discrete Mathematics Demystified
by Steven Krantz
Touches on nearly all of the topics that we hope to cover. We will probably go into greater depth in places, but a very nice and short read.

Fundamental Approach to Discrete Mathematics
by D. P. Acharjya Sreekumar
I also liked this book, however, due to time constraints, this module only focuses on material in chapter 1–4, 8, and 10.



---

*or skim them over a coffee or two.

†I also like *Applied Discrete Structures* by Alan Doerr and Kenneth Levasseur — it is a good source of exercises. (and is free (legally))

Outline

Final Comments on Module

- Discrete Mathematics concepts appear either directly or indirectly in approximately 22 of the 30 modules on your degree.

  $\Longrightarrow$ *Knowing Discrete Mathematics concepts greatly simplfies rest of the course.*

- The module is intended to be an introduction to a large number of topics, so treatment is broad rather then deep.
  - ✔ Most of material is at an introductory level.
  - ⚠ Keeping in sync with material, practicals and tutorials is important.
- The continuous assessment (the practicals) is intended to reenforce the connections between programming and discrete mathematics.

The CA is a "carrot not a stick" — we want you to enjoy the module and keep up to date with the material.