

Logic

# Discrete Mathematics

Number Theory

Topic 00 — Module Introduction

Mathematical Proofs

Lecture 01 — Module Overview

Dr Kieran Murphy   

Recurrence Relations

Department of Computing and Mathematics,  
Waterford IT.  
([kmurphy@wit.ie](mailto:kmurphy@wit.ie))

Set Theory

Autumn Semester, 2021

## Outline

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

Enumeration

1. Module Introduction	2
1.1. <b>What</b> is the Aim of this Module?	3
1.2. <b>Why</b> Study Discrete Mathematics?	13
1.3. <b>How</b> will the Module be Delivered? Assessed?	14
1.4. <b>Who</b> is delivering this module?	17
1.5. <b>When</b> will the module be delivered?	18
1.6. Resources	20

# What?

(Aim)

## Aim, as per 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.

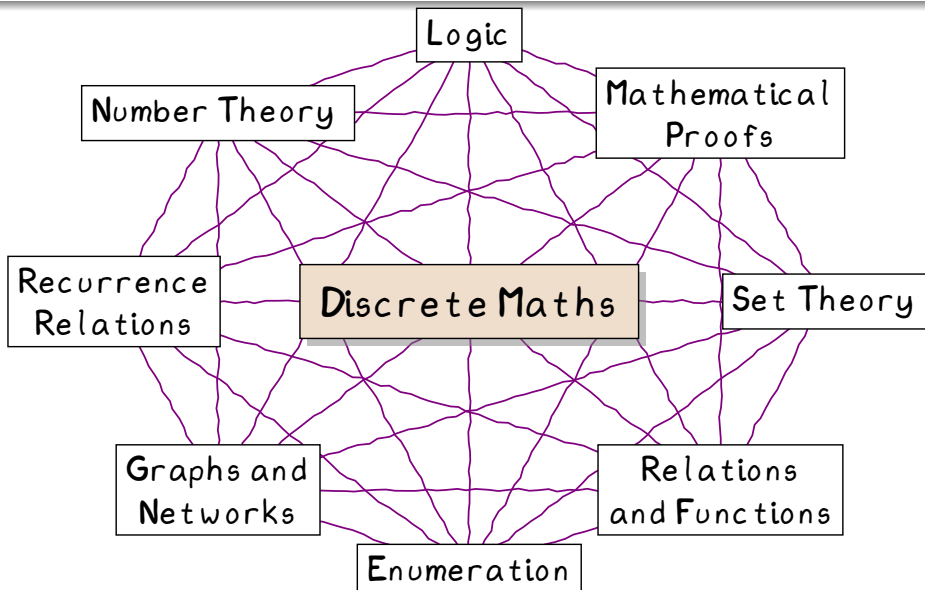
## Translation (Informal Aims)

- 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/languages (Python) ↔ theoretical models (mathematics).

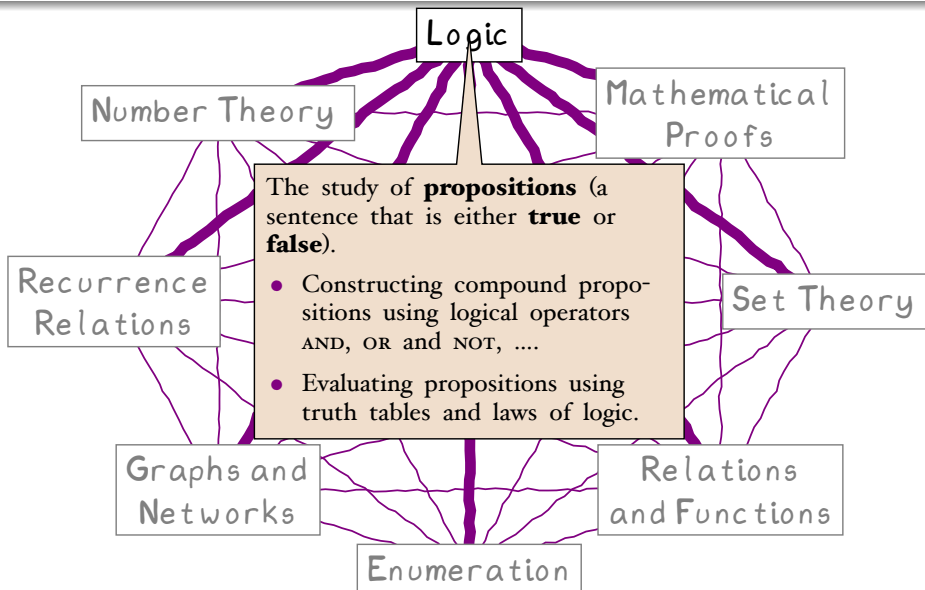
\*Also, see the [module descriptor](#) for the learning outcomes, if that rocks your boat ...

## What?

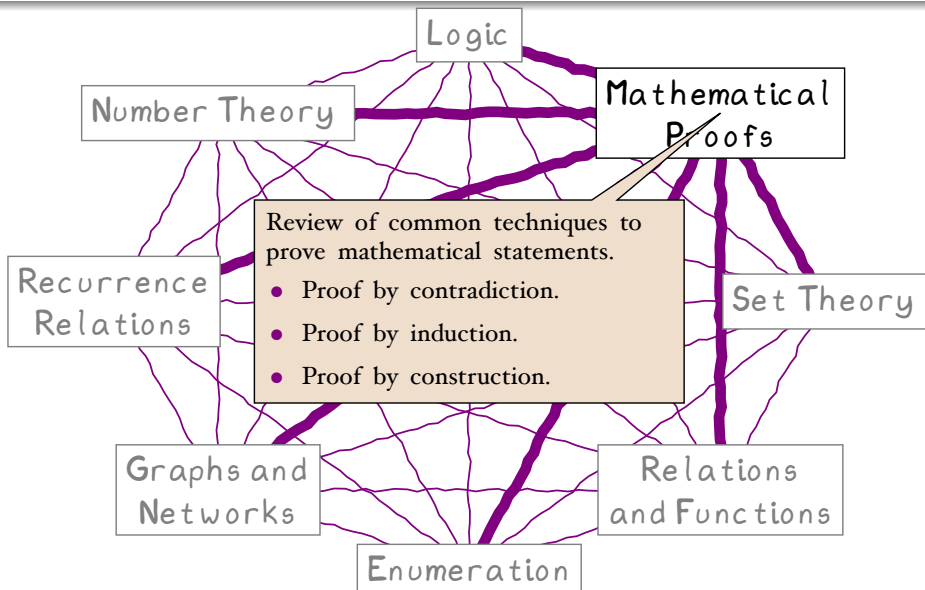
## (Discrete Mathematics)



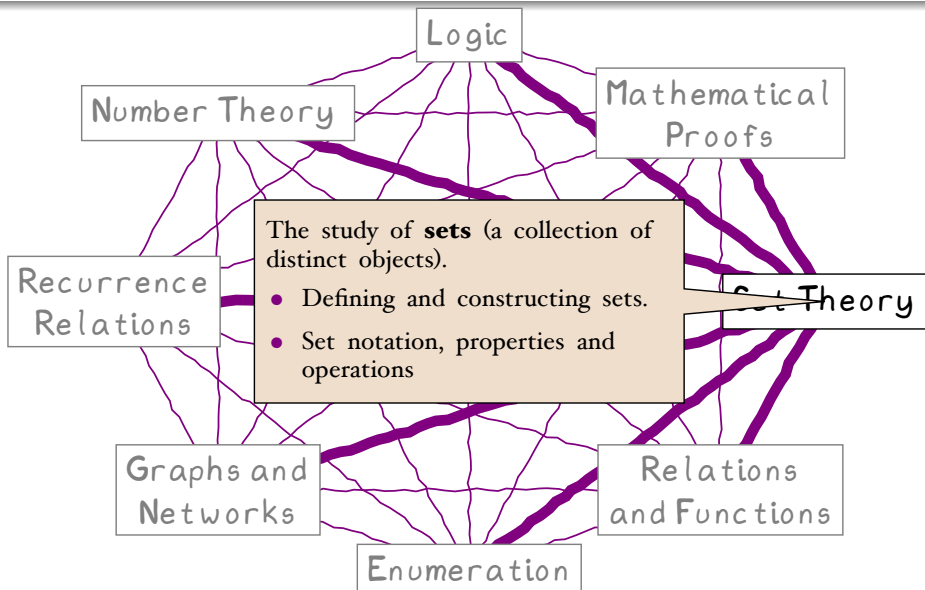
# What?



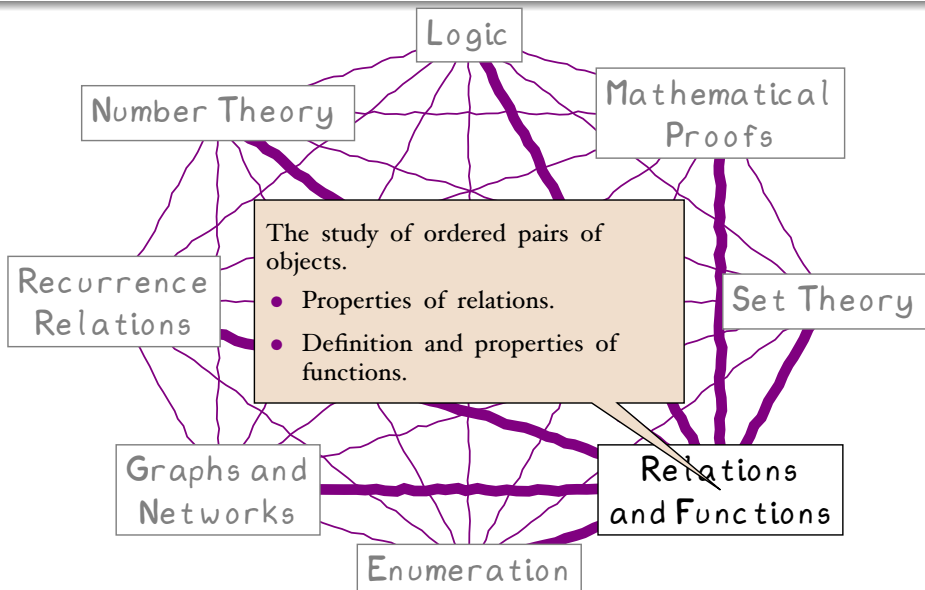
# What?



# What?

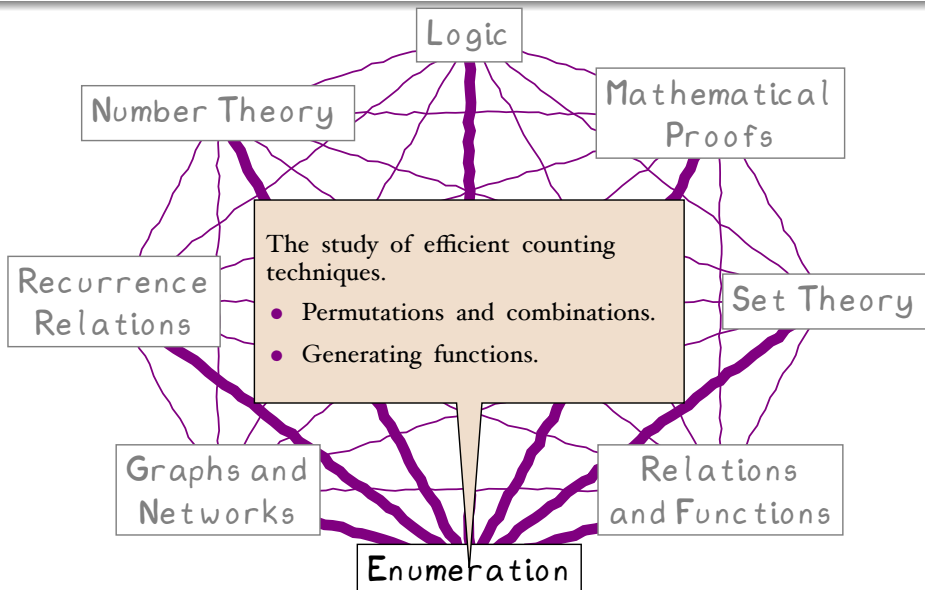


# What?

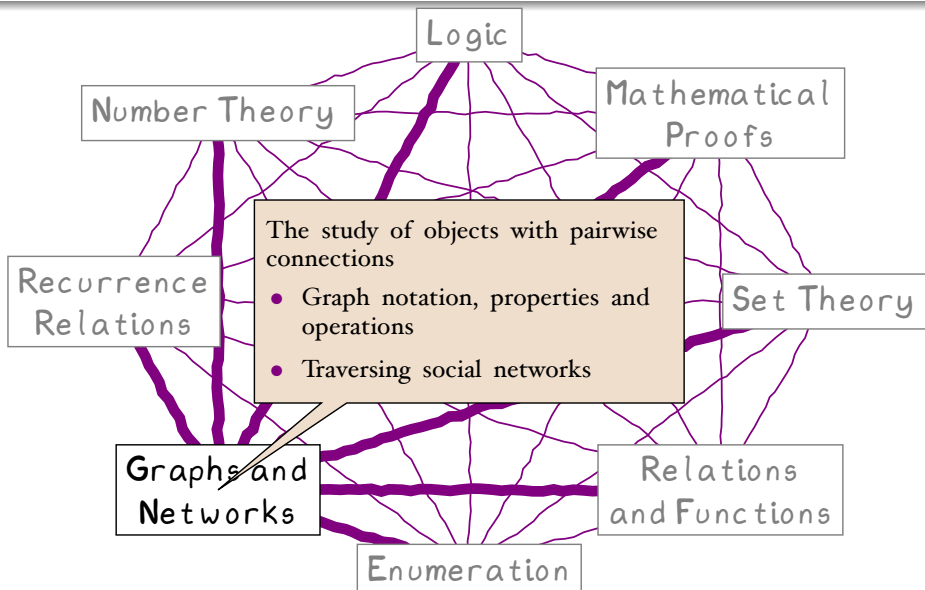




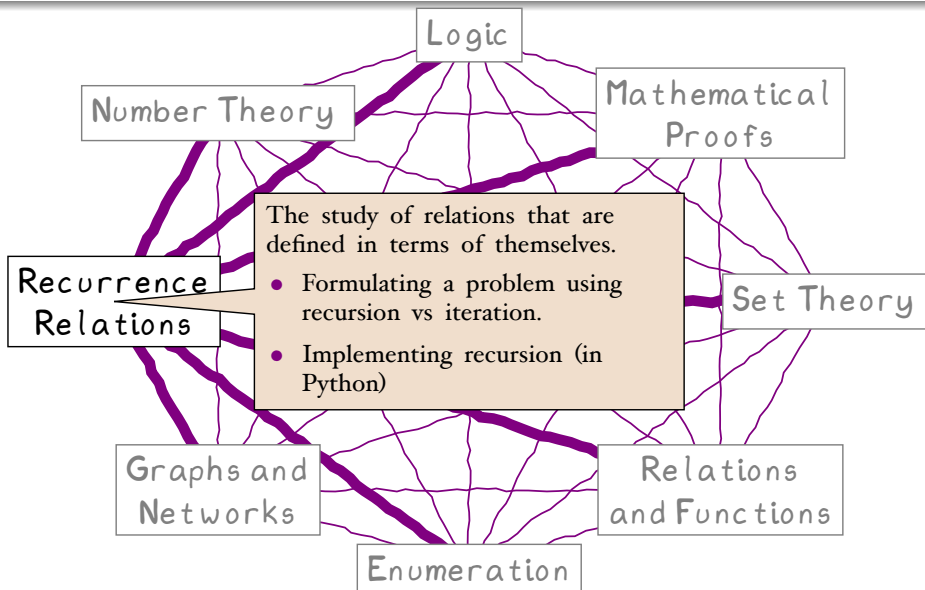
# What?



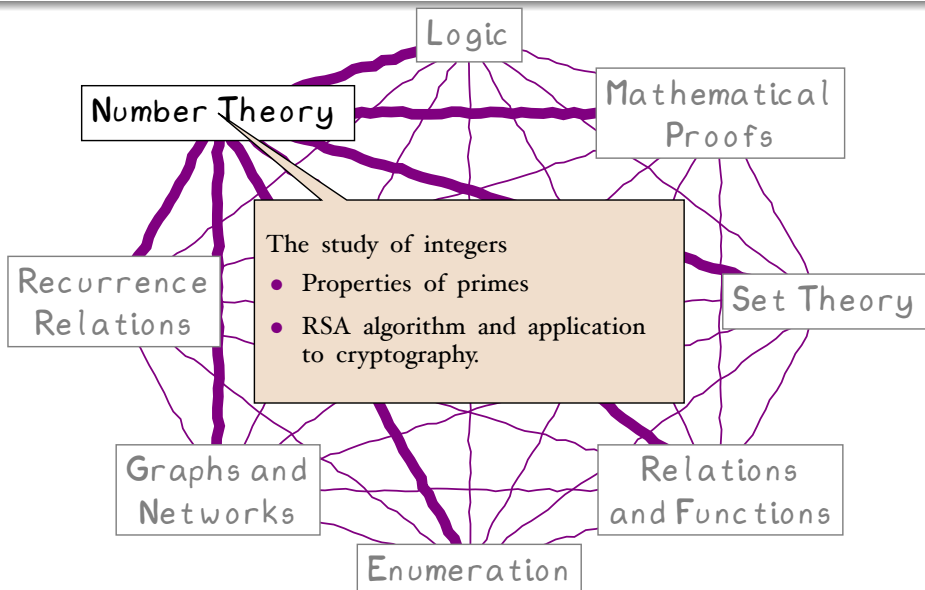
# What?



# What?



# What?



# Why?

Many reasons . . . to pick 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](http://waitbutwhy.com/2015/01/artificial-intelligence-revolution-2.html)

# Why?

Many reasons . . . to pick 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](http://waitbutwhy.com/2015/01/artificial-intelligence-revolution-2.html)

# Why?

Many reasons . . . to pick one . . .

Machine learning is the future of computing

Discrete mathematics is the core of machine learning.

---

\*The AI Revolution: Our Immortality or Extinction

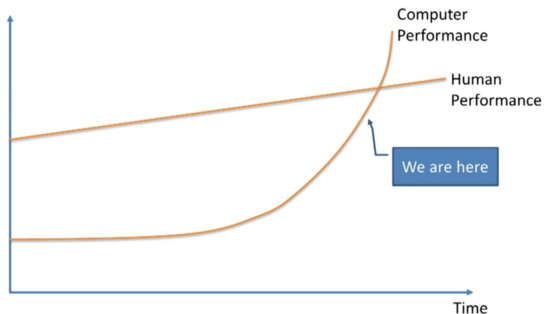
[waitbutwhy.com/2015/01/artificial-intelligence-revolution-2.html](http://waitbutwhy.com/2015/01/artificial-intelligence-revolution-2.html)

# Why?

Many reasons . . . to pick 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](http://waitbutwhy.com/2015/01/artificial-intelligence-revolution-2.html)



# 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 need to have printout of notes in advance of lecture.**
- ⚙️ Ideally you skim over the notes in advance of lecture.

## ⚙️ 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 *jupyter 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?

# (Contact Hours)

## ⚙️ Three lectures per week

- ⚙️ Cover concepts, definitions, examples, etc.
- ⚙️ BUT feel free to stop me and ask questions at any point.
- ⚠️ **You need to have printout of notes in advance of lecture.**
- ⚙️ Ideally you skim over the notes in advance of lecture.

## ⚙️ 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 *jupyter 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?

# (Contact Hours)

## ⚙️ Three lectures per week

- ⚙️ Cover concepts, definitions, examples, etc.
- ⚙️ BUT feel free to stop me and ask questions at any point.
- ⚠️ **You need to have printout of notes in advance of lecture.**
- ⚙️ Ideally you skim over the notes in advance of lecture.

## ⚙️ 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 *jupyter 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.
- 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<sup>†</sup>.
- In theory<sup>‡</sup>, weekly assignments, are graded in advance of next week.

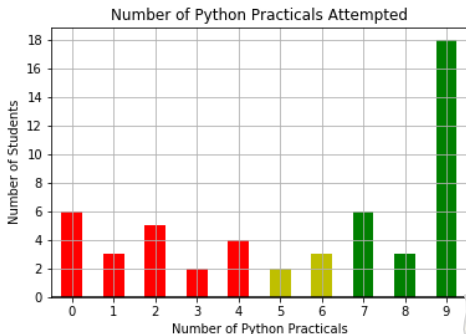
---

<sup>†</sup>The end of topic online quiz are formative assessment only (zero weighting).

<sup>‡</sup>In practice, I ~~may~~ will fall behind a bit.

## A Brief Look at Last Year's Results (2019)

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



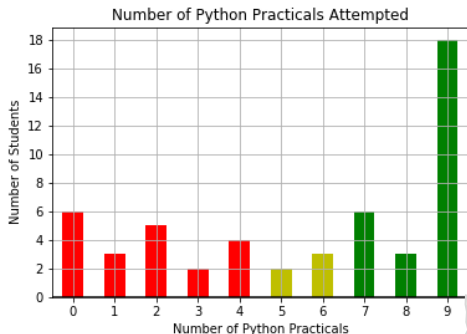
Keep up with the material:

- Read notes before & after lecturers
- Attend practicals
- Attempt tutorial questions.

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

# A Brief Look at Last Year's Results (2019)

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



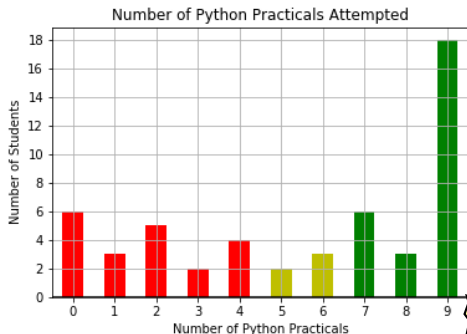
Keep up with the material:

- Read notes before & after lecturers
- Attend practicals
- Attempt tutorial questions.

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

# A Brief Look at Last Year's Results (2019)

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



Keep up with the material:

- Read notes before & after lecturers
- Attend practicals
- Attempt tutorial questions.

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

# Who?

## Dr Denis Flynn

📠 Room 327  
☎ 051-30 2068  
✉ DFlynn@wit.ie



### Background

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

### Academic Interests

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

## Dr Kieran Murphy

📠 Room 313  
☎ 051-30 2055  
✉ KMurphy@wit.ie



### Background

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

### Academic Interests

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



## When?

(Correct as of Sep 21)

KACCM = Applied Computing  
 KCOFO = Computer Forensics and Security  
 KINTT = the Internet of Things

W1 and W2  
 (Denis Flynn)

W3 and W4  
 (Kieran Murphy)

Lectures

TL159 Mon 11:15  
 F26 Tue 10:15  
 F28 Wed 10:15

TL249 Mon 9:15  
 E15 Tue 11:15  
 W14 Fri 12:15

Practicals

W1

W2

W3

W4

IT120 Fri 13:15

TL129 Fri 9:15

IT101 Thu 11:15

IT222 Wed 11:15

Tutorials

TL250 Thu 13:15

FTG23 Thu 9:15

C29 Thu 14:15

IT220 Thu 10:15

# Proposed Calendar<sup>§</sup>

Week	Topic	Practical
0	Intro half-week (1 lecture)	P00 — Introduction to Python and Colab
-----		
1	Logic	P01 — Propositional logic in Python
2	Logic	P02 — Mathematical Proofs in Python
3	Set Theory	P03 — Sets in Python
4	Set Theory	P04 — Applications of Sets
----- Reading Week -----		
5	Relations and Functions	P05 — Implementating Relations in Python
6	Relations and Functions	P06 — Functions and their Properties
7	Enumeration	P07 — Implementing Counting Techniques
8	Enumeration	P08 — Permutations and Combinations
9	Graphs and Networks	P09 — Graph Generation using Networkx
10	Recurrence Relations	P10 — Recursion
-----		
10	Module Review (2 lectures)	P07 — Enumeration in Python
11/12	End of Semester Exams	

<sup>‡</sup>Week 1 starts on Monday 27 September, etc. Please note that this is provisional and will be updated during the semester. In fact, as it stands it is highly unrealistic in predicting that we won't have a break due to Covid, but let's start off being optimistic.

## Resources



- URL: [moodle.wit.ie/course/view.php?id=182312](https://moodle.wit.ie/course/view.php?id=182312)
- Used for all notices, assignment and practical work submissions.



- URL: [DiscreteMathematics-202122.github.io/live](https://DiscreteMathematics-202122.github.io/live)
- Used for all content (slides, notebooks, tutorial sheets).



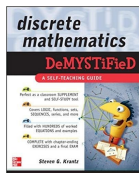
- Used for instant messaging, one-on-one sessions, etc.



We will use the online Google colab environment for python, but if you want to install python on your laptop you could use the anaconda distribution from [www.anaconda.com](https://www.anaconda.com) (just install the 64-bit, version 3.+).

# 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<sup>¶</sup> them<sup>¶</sup>, however, be aware they may use different notation or cover different topics.



## Discrete Mathematics Demystified

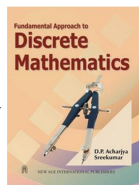
by Steven Krantz

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.




---

<sup>¶</sup>or skim them over a coffee or two.

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

# Final Comments on Module

- Discrete Mathematics concepts appear either directly or indirectly in approximately 22 of the 30 modules on your degree.
  - ⇒ *Knowing Discrete Mathematics concepts greatly simplifies rest of the course.*
- The module is intended to be an introduction to a large number of topics, so treatment is broad rather than 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 reinforce 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.

