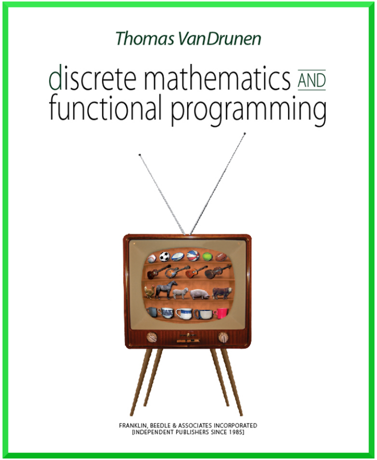
Welcome

CSCI 243
Wheaton College
Thomas VanDrunen
Fall 2025

DMFP

Please keep phones and other devices silenced and put away—not seen or heard (or used) any time during class.



Thomas VanDrunen

discrete mathematics AND functional programming

FRANKLIN, BEEDLE & ASSOCIATES INCORPORATED
[INDEPENDENT PUBLISHERS SINCE 1985]

Discrete Mathematics and Functional Programming.
©2013 Franklin, Beedle and Associates

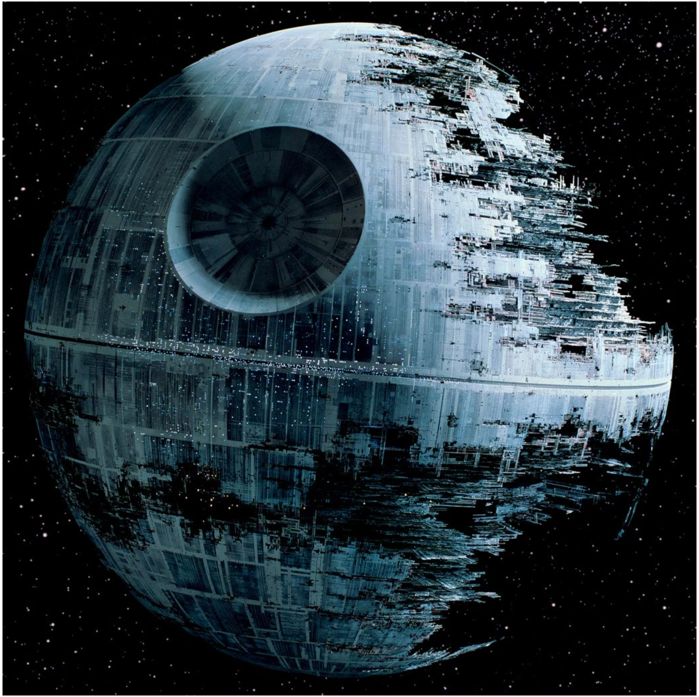
Discrete Mathematics and Functional Programming

with Python

Second Edition

Thomas VanDrunen

Discrete Mathematics and Functional Programming with Python.
Under contract with Taylor & Francis.
Expected 2026.



What is this course about?

Proof-based **discrete mathematics**
and
programming in the **functional style**

. . . with three audiences in mind:

- Computer science majors
- Math majors
- Everyone else

Course outline

Ch 1. Set. (Aug 27–Sept 8)

Ch 2. Sequence. (Sept 10–Sept 19)

Test 1. Sept 24

Ch 3. Proposition. (Sept 26–Oct 3)

Ch 4. Proof. (Oct 6–Oct 13)

Test 2. Oct 17

Ch 5. Relation. (Oct 24–Nov 3)

Ch 6. Function. (Nov 5–Nov 17)

Test 3. Nov 21

Ch 7. Self reference (Nov 24–Dec 10)

Final exam. Tues, Dec 16, 1:30pm

Themes and goals

Throughout this course, we will see these recurring themes:

- Formal definitions
- Recursive thinking
- Analysis and synthesis

Our general goals are that you'll be able to

- M1. Verify mathematical propositions using formats that anticipate mathematical proofs.
- M2. Apply mathematical notation and definitions.
- M3. Write mathematical proofs.
- P1. Use relevant features of the Python programming language correctly.
- P2. Write functions using the Python programming language.
- P3. Reason about the meaning and structure of code.
- R. Think recursively/inductively.

Learning outcome standards

Students will be able to

1. Verify set equality propositions using Venn diagrams. (M1, M2)
2. Use set-builder notation to denote specific sets. (M2)
3. Use Python subscripting on lists and arrays, including slicing, integer-array indexing, and Boolean-array indexing. (P1)
4. Analyze the type of complex Python expressions. (P3)
5. Write Python functions that use set and list comprehensions. (P1, P2)
6. Write Python functions that use recursion. (P2, R; used to assess AAQR 2)
7. Verify logical equivalences using algebraic simplifications. (M1, M2)
8. Verify argument forms using previously known argument forms. (M1, M2)
9. Write Python functions using filter and anonymous functions. (P1, P2)
10. Write Boolean-valued Python functions. (P1, P2, R)
11. Write proofs for basic set propositions [subset, set equality, set emptiness]. (M2, M3; used to assess AAQR 3)
12. Write proofs for advanced set propositions [conditionals, biconditionals, powersets]. (M2, M3)
13. Write proofs for propositions about relations and their properties. (M2, M3)
14. Identify relation properties and derived relations [transitive closure, topological sort]. (M2)
15. Write Python functions that use reduce and related tools. (P1, P2)
16. Write Python functions that process dictionaries. (P1, P2)
17. Write proofs for basic function propositions [image, inverse image, function properties]. (M2, M3)
18. Write proofs for advanced function propositions [interaction among image and function properties, composition] (M2, M3)
19. Write Python functions that operate on recursive types (P1, P2, R; used to assess AAQR 1)
20. Write proofs for propositions about recursively-defined sets using structural induction. (M2, M3, R)
21. Write proofs for loop invariants (M3, P3, R)

Grading

- The gradebook will record whether you have **met (M)**, **progress towards meeting (P)**, or **not met (X)** each standard.
- At Tests 2 and 3 you may retake one standard from a previous test, and at the final exam you may retake as many standards from previous tests as time permits.
- In order to pass this course, your homework/quiz score must be at least 50%; if this score is between 50% and 70%, your letter grade will drop by two steps; between 71% and 85%, one step.

Test 1	Sept 21	Standards 1–6
Test 2	Oct 17	Standards 7–12
Test 3	Nov 21	Standards 13–18
Final exam	Dec 16	Standards 19–21

	Minimum	Minimum at least
Grade	met	progress
A	21	21
A-	19,	21
B+	17	21
B	15	21
B-	13	21
C+	11	18
C	9	18
C-	7	18
D	5	15

How to succeed in this course

- Prepare for class.
- Take the right amount of notes.
- Keep up with the material.
- Ask for help.

5 is a natural number (*or* the collection of natural numbers contains 5). $5 \in \mathbb{N}$

All integers are rational numbers. $\mathbb{Z} \subseteq \mathbb{Q}$

Merging the algebraic numbers and the transcendental numbers makes the real numbers. $\mathbb{R} = \mathbb{A} \cup \mathbb{T}$

Negative integers are both negative and integers. $\mathbb{Z}^- = \mathbb{R}^- \cap \mathbb{Z}$

Transcendental numbers are those real numbers that are not algebraic numbers. $\mathbb{T} = \mathbb{R} - \mathbb{A}$

Nothing is both transcendental and algebraic, *or* the collection of things both transcendental and algebraic is empty. $\mathbb{T} \cap \mathbb{A} = \emptyset$

Adding 0 to the collection of natural numbers makes the collection of whole numbers. $\mathbb{W} = \{0\} \cup \mathbb{N}$

Since all rational numbers are algebraic numbers and all algebraic numbers are real numbers, it follows that all rational numbers are real numbers.

$$\begin{array}{rc} & \mathbb{Q} \subseteq \mathbb{A} \\ & \mathbb{A} \subseteq \mathbb{R} \\ \therefore & \mathbb{Q} \subseteq \mathbb{R} \end{array}$$

For next time:

Read Sections 1.(1 & 2).

Take quiz on Canvas (covering the reading).

Meet in the CSCI lab (MEY 154) on Friday.
This Friday, Aug 29, is the only day we'll meet in the lab.
Make sure you know your username and password for the Wheaton network.