

YORK UNIVERSITY

Lassonde School of Engineering

Department of Electrical Engineering & Computer Science

EECS 1028Z 3.0 — *Discrete Mathematics for Engineers*

Course Outline — Winter 2025

First class: Jan 6, 2025

Instructor	Classes	Tutorials
Dr. George Tourlakis	MWF: 13:30–14:30 Keele VH C	TUT#1: R 17:30/120min, ACW 204
Office: LAS 2051		TUT#2: M 19:30/120min, ACW 305
e-mail: gt@cse.yorku.ca		TUT#3: F 15:30/120min, ACW 305
		TUT#4: F 15:30/120min, ACW 304

The tutorials are mandatory and *may* introduce additional examinable topics.

Students MUST attend the tutorial in which they are registered.

The first class is on Jan 6, 2025 (in person, VH C) and the first tutorial (in person) is in the week of Jan 13, 2025.

Last class is on April 7, 2005.

Reading Week (*no classes, no tutorials*) is Feb. 15–21.

Last date to drop a WINTER 2025 (3-credit) course without receiving a

grade is **March 14, 2025**.

**If you miss this deadline please note
that the Course Withdrawal Period
(i.e., ability to withdraw from a course
and receive a grade of “W” on
transcript) is **March 15 – April 4**.**

Syllabus. This, as the title suggests, is a *first course* in discrete mathematics. That is, the kind of mathematics that people do when they do not want to be bothered with notions of *continuity* and *limits*, the latter being the major concepts and tools in the mathematics of the real and complex numbers (*analysis* or *calculus*).

Here is what we will cover:

- (1) (*Elementary*) **set theory**, which is the basic language that all *users* of mathematics “speak” and “write” (and that includes *computer scientists and engineers* —note, for example, any of the papers, old and new, that have been published over the years in the *Journal of the Association for Computing Machinery*).

We will include here

- Basic definitions and notation (e.g., empty set, unions, intersections, cartesian products).
- An *introduction* to relations and functions and their basic operations and properties.
- An *introduction* to order, equivalence relations and equivalence classes.
- An introduction to cardinality (meaning, quantifying the size of sets) and diagonalisation.
- The *Why and How* of mathematical induction as a tool to prove properties of natural numbers —in proofs **and** “constructions”.
- It is a fact of life that many objects in mathematics, but also in computer science including its applications to engineering are *defined inductively* —or *recursively* as the modern literature prefers to say.

Examples of such objects are formulas, terms, and proofs (in math and in logic), lists, trees, and entire programming languages (in computer science).

In turn, and quite naturally, *properties* of these objects are *argued by induction*.

Thus we will study inductive definitions and *how induction can be employed to prove properties of the objects so defined*. This process will persist and get enriched/refined throughout the course.

(2) **The basic vocabulary and techniques of logic**, including

- propositional calculus, and
- the elementary aspects of predicate calculus.

Here the aim is for us to thoroughly understand the tools for reasoning that the scientist[†] uses (for example, what makes “proof by contradiction” tick?)

(3) **Elementary aspects** of number theory such as

- divisibility
- primes
- *greatest common divisor* and the “extended Euclid’s algorithm”
- notation systems for the integers

These topics prepare the student for work in the analysis of algorithms, as this subject area is dispersed in many sequel courses, and is also found in “concentrated form” in EECS 3101 3.0.

(4) The study of sets, relations and functions will be revisited at a more in-depth manner, including topics such as:

- Closure of relations (transitive closure, reflexive closure, etc.)
- Inverse relations and composition of relations and functions.
- Inductively (recursively) defined *sets* (e.g., sets of formulas of logic, sets of trees) as *closures*.
- Structural Induction.

Prerequisite

MHF4U and MCV4U.

You will *not* need to know any computer programming, but, of course, a broader background always helps!

Course work and assessment:

There are **mandatory** 2-hour long tutorials for this course (**once a week**). At the top of this outline you will find the schedule that *pertains to the tutorial you registered in*.

[†]All engineers are scientists, but not the other way around.

- (A) **Homework:** *At least 4 Problem Sets* 28%

The homework must be each individual's own work. While consultations with the instructor, tutor, and among students, are part of the learning process and are encouraged, nevertheless, at the end of all this consultation **each student will have to produce an individual report** rather than a copy (full or partial) of somebody else's report.



The concept of “late assignments” does not exist in this course (*because* full solutions are posted on the due date).



Policies.

It is important to follow these links to familiarise yourselves with Senate's and Lassonde School's Policies and expectations regarding Academic Honesty, and Academic Integrity.

Please also familiarise yourselves with these Senate Policies:

Academic Accommodation for Students with Disabilities, Religious Observance (acommodation), Repeating Passed or Failed Courses for Academic Credit.

But also check these two links! *Student Rights and Responsibilities* and *Student Accessibility Services*.

- (B) **Mid-Term Exam** (*In March (TBA)*) (60 minutes, **in person**) 32%



Missed tests with good reason (normally medical, and well documented) will have their *weight transferred* to the final exam. *There are no “make up” tests.* **Tests missed for no acceptable reason are deemed to have been written and failed and are graded “0” (F).** *There are no “make up” assignments.*

The only time the weight of an assessed component is transferred to the final is when the component **is missed** with due cause (illness). **This does not apply to assignments** since the student has typically 2–3 weeks to do any given assignment.



- (C) **Final Exam** during the University's Exam period. 40%

Textbook.

George Tournakis, *Discrete Mathematics; A Concise Introduction*.

ISBN 978-3-031-30487-3, Springer-Nature, Switzerland, 2024.

The chapters in our text as well as our “slides” support learning the material covered in class and the *learning outcomes* of the course:

Course Learning Outcomes (CLO)s: On successful completion of the course, students will be able to:

- CLO 1.** Specify the domain and range of a given function, as well as the left and right fields (1d)
- CLO 2.** Use the cardinality concept and compute cardinalities of finite and infinite sets. (1f)
- CLO 3.** Use Boolean or predicate logic to establish the truth of a mathematical statement (2a)
- CLO 4.** Contrast truth table and syntactic proof techniques in the proof of tautology. (2f)
- CLO 5.** Prove simple mathematical statements by using Boolean or predicate logic, as needed. (2e)
- CLO 6.** Be able to reason about trees and other inductively defined sets using structural induction. (3e)