EXPANDED COURSE DESCRIPTION
Lassonde School of Engineering
Electrical Engineering Computer Science
LE / MATH 1028 3.0 SECTION M
DISCRETE MATHEMATICS FOR ENGINEERS
FALL 2019 / WINTER 2020

Last Modified Date: 12/17/2019

COURSE CALENDAR DESCRIPTION

An introduction to propositional logic and application to switching circuits; sets, relations and functions; predicate logic and proof techniques; induction with applications to program correctness; basic counting techniques with applications; graphs and trees with applications in circuit analysis, information storage and retrieval, Huffman coding; automata and applications in software engineering. Prerequisites: MHF4U (Advanced Function) and MCV4U (Calculus and Vectors). Course Credit exclusions: LE/CSE 1019 3.00 (prior to Fall 2014), LE/EECS 1019 3.00, SC/CSE 1019 3.00 (prior to Summer 2013), SC/MATH 1019 3.00, SC/MATH 2320 3.00.

Course Listed Courses: EECS 1028

INSTRUCTOR(S)

<table>
<thead>
<tr>
<th>Name</th>
<th>Section / Format / Term</th>
<th>Contact Email</th>
<th>Contact Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tourlakis, George</td>
<td>Sec. M / LECT / W</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ADDITIONAL INFORMATION

The tutorials are mandatory and may introduce additional examinable topics. Students will attend the tutorial in which they are registered.

The first class is on Jan. 8, 2020 and the first tutorial is in the week of Jan. 13, 2020.

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

Last date to drop a Winter 2020 (3-credit) course without receiving a grade is March 13, 2020. 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 14-April 5.

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
   • “floors” and “ceilings”
   • 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.

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

4. “Big-O”-notation
   • An introduction to recurrence relations, along with techniques to solve them, such as the method of generating functions.

This topic prepares for work in the analysis of algorithms, in particular it relates to the technique of “divide and conquer” of which we will see at least two instances (binary search, two-way-merge-sort).

5. 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.
   • Rooted trees.
   • Counting trees that have certain properties and the relevance to algorithm analysis.

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). See at the top of this outline the schedule that pertains to the tutorial you registered in.

Homework
At least 4 Problem Sets - 20%
   • 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).

Mid-Term Exam
Mid-Term Exam (in-class) - 40%

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.

**Final Exam**

Final Exam during the University’s Exam period - 40%

**Textbook**


The following chapters from the text support learning the material covered in class:

*Ch. 1-2, Sections 3.3 and onwards in Ch. 3, Ch. 4, 8, 9, 11.*

**Course Learning Objectives**

Students are expected to:

- **CLO1.** Prove or disprove any propositional formula as the case requires, using truth tables or syntactic proof techniques such as resolution.
- **CLO2.** Prove or disprove as the case may be simple formulas in quantified logic.
- **CLO3.** Translate English mathematical statements into predicate logic formulas.
- **CLO4.** Prove simple mathematical statements by contradiction, by cases, or by assuming the antecedent.
- **CLO5.** Prove by induction statements that depend on a natural number; in particular: Prove the correctness of single loop programs and simple recursive programs.
- **CLO6.** Prove statements about inductively defined objects by structural induction; in particular: Prove the correctness of simple recursive programs.
- **CLO7.** Be able to reason about graphs and (binary) trees and use them in several examples, e.g., demonstrate an ability to locate the fundamental cycles of an electrical circuit
- **CLO8.** Be able to show simple properties of trees (examples: relation be- tween number of nodes and edges; relation between number of nodes and height)

**ACADEMIC INTEGRITY LINKS**

- Senate Policy on Academic Honesty - [http://secretariat-policies.info.yorku.ca/policies/academic-honesty-senate-policy-on/](http://secretariat-policies.info.yorku.ca/policies/academic-honesty-senate-policy-on/)
- Academic Integrity - [http://lassonde.yorku.ca/academic-integrity](http://lassonde.yorku.ca/academic-integrity)

**STUDENT LINKS**

- Student Rights and Responsibilities - [http://oscr.students.uit.yorku.ca/student-conduct](http://oscr.students.uit.yorku.ca/student-conduct)
- Religious Observance - [https://w2prod.sis.yorku.ca/Apps/WebObjects/cdm.woa/wa/regobs](https://w2prod.sis.yorku.ca/Apps/WebObjects/cdm.woa/wa/regobs)
- Student Accessibility Services (SAS) - [https://accessibility.students.yorku.ca/](https://accessibility.students.yorku.ca/)
- York University’s Policies on Gender/LGBTQ*/Positive Space - [http://rights.info.yorku.ca/lgbtq/](http://rights.info.yorku.ca/lgbtq/)

**LAND ACKNOWLEDGEMENT**

- We acknowledge our presence on the traditional territory of many Indigenous Nations. The area known as Tkaronto has been care taken by the Anishinabek Nation, the Haudenosaunee Confederacy, the Huron-Wendat, and the Métis. It is now home to many Indigenous Peoples. We acknowledge the current treaty holders, the Mississaugas of the New Credit First Nation. This territory is subject of the Dish With One Spoon Wampum Belt Covenant, an agreement to peaceably share and care for the Great Lakes region.
• The Indigenous Framework for York University: A Guide to Action can be found here:
  http://indigenous.info.yorku.ca/
• Meaning of a land acknowledgement: http://healthydebate.ca/opinions/indigenous-land-acknowledgements

Many courses utilize Moodle, York University’s course website system. If your course is using Moodle, click here to access it.
Moodle @ York University