McMaster University
Department of Computing and Software
Dr. W. Kahl

COMPSCI&SFWRENG 2DM3
Outline
2017-09-05

Discrete Mathematics and Applications 1

COMPSCI&SFWRENG 2DM3 — Fall 2017

Instructor: Dr. Wolfram **Kahl**, Department of Computing and Software, ITB-245
E-Mail: kahl@cas.mcmaster.ca
Office Hours: *Preliminary*: Wed. 13:00–14:00, and by appointment.

Calendar Description:

Functions, relations and sets; the language of predicate logic, propositional logic; proof techniques, counting principles; induction and recursion, discrete probabilities, graphs, and their application to computing.

Goals:

This course will teach the mathematical foundations of computer science.

To a large degree, this can be seen as analogous to acquiring **language skills**, including **knowledge** and **skills** concerning syntax, semantics, pragmatics, and vocabulary of the **language of (discrete) mathematics**.

Conscious and precise use of this language is the foundation for **precise specification and rigorous reasoning**, which take a central place in this course.

Course Page: `http://www.cas.mcmaster.ca/~kahl/CS2DM3/2017/`

While most of the internal electronic information exchange for this course will be handled via **Avenue**, the course pages will contain useful links to external material, and will also serve as central fallback location for making information and material available outside Avenue, in particular in the case of Avenue accessibility problems. It is the student's responsibility to be aware of the information on the course Avenue site and on the course web page, and to check regularly for announcements (or RSS subscribe, where possible).

Schedule:

Lectures: Tuesday, Wednesday, Friday 15:30–16:20, CNH-104

Tutorials:

T1	Mo	12:30–13:20	ETB-119	T5	We	9:30–10:20	BSB-117
T2	Mo	9:30–10:20	ETB-119	T6	We	8:30–9:20	ETB-237
T3	We	13:30–14:20	ETB-237	T7	Mo	14:30–15:20	CNH-102
T4	Tu	13:30–14:20	ETB-237	T8	Mo	15:30–16:20	BSB-105

Tutorials start on the 11th of September.

Occasionally, special sessions in computer-equipped labs may be announced on Avenue and course pages.

Students are expected to attend all lectures and tutorials.

Textbook

"LADM": David Gries and Fred B. Schneider: **A Logical Approach to Discrete Math**, Springer 1993, ISBN 3-540-94115-0

> "This is a rather extraordinary book, and deserves to be read by everyone involved in computer science and — perhaps more importantly — software engineering. I recommend it highly [...] . If the book is taken seriously, the rigor that it unfolds and the clarity of its concepts could have a significant impact on the way in which software is conceived and developed."
>
> — Peter G. Neumann

Additional material will be made available electronically via the course pages.

Outline:

(With the most relevant textbook chapters indicated — not all textbook contents will be covered in detail, and material will be interleaved heavily. Times are rough estimates.)

• Introduction to Calculational Reasoning	Parts of Chapters 1, 15	≈ 2 weeks
• Boolean Expressions and Propositional Logic	Chapters 1–5	≈ 2 weeks
• Syntax and Semantics of Predicate Logic	Chapters 8–9	≈ 1.5 weeks
• Predicates and Programming	Chapter 10	≈ 1 week
• Sets	Chapter 11	≈ 1 week
• Relations and Functions	Chapter 14	≈ 2 weeks
• Induction and Sequences	Chapters 12–13	≈ 1.5 weeks
• Graphs, Counting	Chapters 19, 16	≈ 1.5 weeks

Using **tool support** will be part of the expectations; details to be announced.

Exercises and Tutorials:

In most weeks, **Exercises** will be provided, which will constitute the main material for the tutorials. Additionally, **Assignment Questions** will be provided, which will usually be due on Monday mornings before the first tutorial,

Every week, starting September 11, there will be one-hour tutorial session in eight tutorial groups. The main purpose of the tutorials is to discuss **student work** on exercise problems. Therefore, every student is expected to complete the scheduled work, i.e., exercise problems or necessary reading, **before** the corresponding tutorial session — in particular, solutions and solution attempts to the Exercises of the current week are to be brought to the tutorial.

Since space in the tutorial rooms is limited, the TAs are under strict guidelines to ask students not registered for the currently-scheduled tutorial group to leave if the tutorial room is overcrowded.

Grading:

All examinations in this course will be **Closed Book**. That is, no written or printed material nor a calculator nor other electronic aids except for the tools specified for use in each exam may be used during the examinations.

> After notations, presentation rules, and basic definitions and proof rules have been introduced in class, **students are expected to know them at all times**.

Assignments: There will be graded **Assignment Questions**, and ungraded **Exercises**, essentially on a weekly basis.

Assignments may be graded only automatically and/or summarily; evaluation will be conducted mostly via the midterm tests and the final.

It is essential that you meet the deadlines for the Assigment Questions; there is no credit for documents handed in or files submitted after the deadline.

Midterms: There will be **three midterm examinations**. These will be written **in the UTS computer labs using CalcCheck tool support** on the computers. Due to lab availability, the midterms will be written within the following time slots:

Midterm 1:	Friday, 29th September	17:45 to 19:15
Midterm 2:	Friday, 27th October	17:45 to 19:15
Midterm 3:	Friday, 24th November	17:45 to 19:15

You will receive an e-mail specifying your lab and seat number the night before each midterm.

Final Exam: The **final examination** will be scheduled by the Registrar's Office in the usual way. It will be a closed book examination of 2.5 hours (150 minutes) duration and cover the material of **all** lectures, tutorials, handouts, and assignments. The current hope is that the final exam will be written in computer labs using tool support on the computers, but it may turn out that this will not be possible due to organisational reasons beyond the control of the instructor.

Grade Calculation: All exam grades will be percentage grades.

For every student, the course grade is calculated as a weighted average:

- For n being the number of **assignments** this course will have had, 10% of the weight are given to your $(n-2)$ best assignments. This allows you to miss 2 assignments — there will be **no further accommodations for missed assignments**. (***In particular, MSAFs for assignments will not be processed.***)

  Some exercise sheets or assignments may contain **bonus questions**. All bonus marks will be added to the course grade *only for those who have passed the course otherwise.*

- Those **midterms** where your result is better than your result in the final count 20% each, and those midterms that are not better than the final count 10% each.
- The remaining weight (between 30% and 60%) is given to the **final exam**.

The final course grade will be converted from a percentage grade to a letter grade according to the scale of the Registrar's Office.

The instructor reserves the right to conduct any deferred midterm or final exams orally.

Course Adaptation

The instructor and university reserve the right to modify elements of the course during the term.

The university may change the dates and deadlines for any or all courses in extreme circumstances. If either type of modification becomes necessary, reasonable notice and communication with the students will be given with explanation and the opportunity to comment on changes.

It is the responsibility of the student to check their McMaster email and course websites weekly during the term and to note any changes.

> The current plan is to have all midterms and the final written on computers using CalcCheck tool support. It is not yet certain whether this will be feasible for the final, as mentioned above. In addition, in case of system/network outages, exams written in the computer labs may still need to be (partially) written by hand.
>
> Therefore, you need to be both **fluent in producing formalisations and proofs using the CalcCheck tool** *and* **fluent in writing syntactically correct formalisations and proofs by hand**.

Academic Ethics

You are expected to exhibit honesty and use ethical behaviour in all aspects of the learning process. Academic credentials you earn are rooted in principles of honesty and academic integrity.

Academic dishonesty is to knowingly act or fail to act in a way that results or could result in unearned academic credit or advantage. This behaviour can result in serious consequences, e.g. the grade of zero on an assignment, loss of credit with a notation on the transcript (notation reads: "Grade of F assigned for academic dishonesty"), and/or suspension or expulsion from the university.

It is your responsibility to understand what constitutes academic dishonesty. For information on the various types of academic dishonesty please refer to the Academic Integrity Policy, located at `http://www.mcmaster.ca/academicintegrity`.

The following illustrates only four forms of academic dishonesty:

1. Plagiarism, e.g. **the submission of work that is not one's own** or for which other credit has been obtained.
2. **Collaboration where individual work is expected.**

   > **You have to produce your submissions for assignment questions yourself, and without collaboration.**
   >
   > For each assignment question there will normally be exercise questions similar to it — you **are allowed** to collaborate on these **exercise questions**. (The tutorials are typically not expected to cover all exercise questions.)

3. Improper collaboration in group work.
4. **Copying or using unauthorised aids in tests and examinations.**

Academic Accommodation of Students with Disabilities

Students who require academic accommodation must contact Student Accessibility Services (SAS) to make arrangements with a Program Coordinator. Academic accommodations must be arranged for each term of study. Student Accessibility Services can be contacted by phone 905-525-9140 ext. 28652 or e-mail sas@mcmaster.ca. For further information, consult McMaster University's Policy for Academic Accommodation of Students with Disabilities.

Discrimination

The Faculty of Engineering is concerned with ensuring an environment that is free of all adverse discrimination. If there is a problem that cannot be resolved by discussion among the persons concerned, individuals are reminded that they should contact the Department Chair, the Sexual Harassment Office or the Human Rights Consultant, as soon as possible.

Use of Avenue

In this course we will be using "Avenue to Learn". Students should be aware that, when they access the electronic components of this course, private information such as first and last names, user names for the McMaster e-mail accounts, and program affiliation may become apparent to all other students in the same course. The available information is dependent on the technology used. Continuation in this course will be deemed consent to this disclosure. If you have any questions or concerns about such disclosure please discuss this with the course instructor.

Learning Objectives

Course Precondition

Students are expected to have achieved the following learning objectives before taking this course:

1. Students should know and understand
   a) Basic number types, such as integers, rationals, reals
   b) General principles of mathematical notation
   c) General principles of calculation in mathematics
   d) Basic imperative programming
2. Students should be able to
   a) Perform calculations and solve equations in numerical domains
   b) Write and "mentally execute" simple imperative programs

Course Postcondition

Students are expected to achieve the following learning objectives **as a result of taking this course**:

1. Students should know and understand
   a) Syntax of propositional logic
   b) Syntax of predicate logic, including variable binding issues
   c) Principles of typed expressions, and the types of the operators they are using
   d) Basic semantics of expressions and formulae (truth tables, validity, ...)
   e) Principles of calculational proofs
   f) Basic and derived proof rules of propositional logic
   g) Basics of (typed) set theory
   h) Basic concepts and theorems about functions and (especially binary) relations
   i) Basic theory of integers and counting
   j) Basic graph theory
2. Students should be able to
   a) Extract the propositional-logic structure from English sentences, and translate propositional expressions back into English
   b) Translate English statements of moderate complexity into predicate-logic specification
   c) Translate simple mathematical prose into predicate-logic definitions and formulae
   d) Produce calculational proofs in propositional logic
   e) Explain and prove basic properties using induction
   f) Produce calculational proofs in applications to set and relation theory, integers and counting, etc.
   g) Explain and prove basic graph properties.

Cooverage of Graduate Attribute Indicators by Learning Outcomes (Preliminary)

For each indicator, the learning outcome topics (defined in the rubrics below) used to measure it are indicated:

1 (A knowledge base for engineering)

1.1 (Competence in Mathematics)

- Logical Formalism
- Calculational Proofs in Propositional Logic
- Calculational Proofs in Predicate Logic Applications
- Induction
- Formalisation
- Discrete Structures: Sets, Functions, Relations, Graphs, Counting

Learning Outcomes Rubric (Preliminary)

Topic	**Below**	**Marginal**	**Meets**	**Exceeds**
Logical Formalism	when writing formulae, frequently introduces syntax or type errors	writes mostly syntactically correct formulae with occasional type errors	writes syntactically and type-correct formulae, and correctly renames variables when substituting	can explain typing and variable binding issues with good understanding
Calculational Proofs in Propositional Logic	cannot produce correct calculational proofs even for simple theorems	typically makes about one mistake every three proof steps	normally produces correct proof calculations, but may not succeed with more complex tasks, or take unnecessary detours	confidently produces even more complex propositional-logic proofs
Calculational Proofs in Predicate Logic Applications	cannot produce correct calculational proof steps using predicate-logic proof rules even in simple contexts	shows some ability to apply predicate-logic proof rules in simple contexts	correctly applies predicate-logic proof rules in simple contexts, and can at least attempt to tackle more complex settings	correctly applies predicate-logic proof rules also in more complex settings
Induction	cannot handle even simple induction proofs	can mostly produce simple routine induction proofs	confidently performs simple and nested induction proofs	handles even complex induction settings confidently
Formalisation	cannot handle even simple formalisation tasks	can produce simple formalisations that mostly capture the natural-language meaning	confidently performs simple formalisation tasks, but may not be able to cope with more complex tasks	produces reasonable formalisations even for moderately more complex tasks
Discrete Structures: Sets, Functions, Relations, Graphs, Counting	demonstrates hardly any understanding of basic concepts of discrete structures	knows and can apply some basic concepts of discrete structures	knows and can apply the majority of the concepts of discrete structures taught in class	demonstrates a solid understanding of the concepts of discrete structures taught in class