Formal Methods in Software Engineering (CISC/CMPE 422/835):

Syllabus

Juergen Dingel
School of Computing
Queen's University

September 5, 2023

1 Course and Lecture Information

Course term: Fall 2023
Delivery format: In person
Course web page: Hosted in OnQ. To access, use your Queen's credentials to log into OnQ via onq.queensu.ca
Times and locations of lectures: See Solus.

2 Teaching Staff Information

Instructor: Juergen Dingel
Web page: www.cs.queensu.ca/~dingel
Email: dingel@queensu.ca
Office hours: See OnQ
Teaching assistants:
See OnQ

3 Course Outline

Modern software development inevitably requires the design and analysis of a number of different artifacts (e.g., requirements documents, design documents, documentation, code, test suites, user guides, build files, deployment scripts). Formal methods allow the mathematically precise formulation of some of these artifacts. For instance, formulas in propositional or predicate logic can capture operational requirements, state machines can describe the behaviour of code fragments and protocols, and class models can capture static designs. The advantage of using these formal notations is that they typically improve the overall quality of the artifacts by removing ambiguities and imprecisions, and enabling automatic analyses that can establish desirable properties or uncover undesirable properties. Consequently, the use of formal methods is indicated in domains in which the software has to meet very high quality standards and failure cannot be tolerated such as air-traffic control. Moreover, the abstraction and automation capabilities of some formal techniques present a powerful weapon against the ever-increasing complexity of software.

CISC/CMPE 422/835 is an introduction to the use of formal methods for the specification, design, and automatic analysis of software systems. The course will present a variety of specification notations (e.g., propositional and predicate logic, Alloy, UML/OCL, temporal logic), and discuss corresponding analysis techniques (theorem proving, constraint checking, animation, testing, and model checking) using existing, state-of-the-art tools (e.g., Jape, Alloy, USE, Jqwik, and NuSMV). The course is most suited for students with a general background in computer science or computer engineering and in interest in the theory and practise of software development and/or modeling and analysis.

CISC/CMPE 422/835 complements the theoretical material with exposure to many practical, industrially-relevant software development concepts, techniques and tools including software repositories such as GitHub, build automation tools such as Gradle, parser generators such as ANTLR, garbage collection, concurrent programming in Java, and barrier synchronization.

4 Intended Student Learning Outcomes

To complete CISC/CMPE 422 students will demonstrate their ability to

1. use and explain formal specification languages based on, e.g., propositional logic, predicate logic, relational calculus, and finite state machines,
2. use and explain notations and techniques to define the semantics of a language precisely,
3. use and explain analysis techniques for formal specification languages such as theorem proving, satisfiability checking, automatic test input generation, and exhaustive state space exploration together with their capabilities and limitations,
4. use and explain tools supporting formal specification languages together with their capabilities and limitations,
5. design, construct, and analyze small formal specifications, and
6. explain the advantages and disadvantages of formal specification languages and tools, and
7. explain the role and potential uses of formal methods for different software development activities.

Students in CISC 835 will additionally

1. demonstrate potential for independent research and development via a course project,
2. describe the process and results of their independent project in a written report, and
3. summarize the motivation and key results of their independent project in an oral presentation.

5 Assessment of Learning Outcomes

The degree to which students have achieved these outcomes will be assessed using a variety of ways of which assignments, midterms, and participation in student engagement activities will be the most important (see the grading scheme below).

6 Textbooks and Readings

The course has a courseware package that will be made available electronically to registered students in OnQ. Unless explicitly stated otherwise, this package is required reading. There is no required text book. However, students looking for supplemental reading might find the following useful:

> M. Huth and M. Ryan. *Logic in Computer Science: Modeling and Reasoning about Systems.* Cambridge University Press. 2nd Edition. 2004.

The book also has a web page with a WWW tutor. Additional material such as sample specifications used in class will be made available via OnQ.

7 Grading Scheme and Grading Method

The grade of students in CISC/CMPE 422 will be computed from their grades in the following 6 deliverables using the indicated weights.

- 4 assignments (40%) where
  - Assignment 1: 10%
  - Assignment 2: 10%
  - Assignment 3: 10%
  - Assignment 4: 10%
- 1 mid-term exam (20%)
- 1 final exam (40%)

For students in CISC 835, the following scheme will be used:

- 4 assignments (30%) where
  - Assignment 1: 7.5%
  - Assignment 2: 7.5%
  - Assignment 3: 7.5%
  - Assignment 4: 7.5%
- 1 mid-term exam (15%)
- 1 final exam (35%)
- course project (20%):
  - proposal (2 pages) and final report (at least 5 pages): 6%
  - presentation (20 mins): 7%
  - project evaluation, i.e., overall difficulty and quality of work carried out: 7%

All components of this course will receive numerical percentage marks. The final grade you receive for the course will be derived by converting your numerical course average to a letter grade according to Queen's Official Grade Conversion Scale:

Grade	Numerical Course Average (Range)
A+	90–100
A	85–89
A−	80–84
B+	77–79
B	73–76
B−	70–72
C+	67–69
C	63–66
C−	60–62
D+	57–59
D	53–56
D−	50–52
F	49 and below

8 Assignments

8.1 All Assignments are Individual

All assignments in the course are *individual* assignments. That means every student in the course must prepare and submit their own set of answers as specified in the assignment description. Students may discuss general approaches to solve a problem, but not the specifics. For instance, it is *not allowed* for students to give other students access to their answers, even if these answers are incomplete or preliminary. Doing so counts as *facilitation* and as a violation of academic integrity. Conversely, students who obtain answers from elsewhere and either submit these or use them to prepare a submission commit plagiarism. For more information on academic integrity, see `www.cs.queensu.ca/undergraduate/syllabus/2023-24.php#academic-integrity`.

8.2 Assignment Distribution

Assignment descriptions and starter code or models (if any) will be distributed via GitHub Classroom. Every student in the course will need an account on GitHub so that they can create their own local repository containing description and starter material. More detailed instructions including a very short introduction to working with GitHub will be part of each assignment.

8.3 Assignment Submission

Solutions to assignments need to submitted to *and*. For GitHub, the last commit pushed to the master branch before the assignment's deadline will be considered your final submission. For the OnQ submission, download an archive of your repository from GitHub and upload it as is to OnQ. Only submissions present on both GitHub and OnQ will be graded. Again, more detailed submission instructions will be part of every assignment.

9 Tentative Schedule of Assignments and Midterms

Assignments and midterms are subject to the following tentative dates (exact dates will be finalized by the end of Week 2 the latest).

	Topic	Out	Due
Assignment 1	Evaluating predicate logic queries for time series data	Thurs, Sept 7	Thurs, Sept 28
Assignment 2	Modeling decision trees w/ Alloy	Thurs, Sept 28	Tue, Oct 24
Assignment 3	Property-based testing w/ Jqwik	Tue, Oct 24	Fri, Nov 10
Assignment 4	Reasoning about barrier synchronization w/ NuSMV	Fri, Nov 10	Tue, Dec 5
Midterm	Alloy	Fri, Nov 3	
Final exam	All course material	Tba	

Assignments will typically be due at 5pm on the indicated dates.

10 Late Policy

There is no late policy for assignment submission. Submissions after the stated due date of an assignment will not be accepted.

11 Generative AI tools

The use of generative AI tools (large language models) such as ChatGPT on assignments is permitted. However, the way it has been used must be described. I.e., in a separate document the questions that the AI tool was used for and how must be summarized, possibly together with sample prompts (if any). The description can be brief (up to 0.5 page, excluding prompts.

12 Data Sheet

All students are allowed to use one data sheet (size: 8.5 by 11 inches) with information on both sides (no flaps, no tricks, etc) for the midterm and the final exam. No other aids are allowed.

13 Calculator Policy

Calculators are not allowed during midterms or exams.

14 Turnitin Statement

This course makes use of Turnitin, a third-party application that helps maintain standards of excellence in academic integrity. Normally, students will be required to submit their course assignments to through onQ to Turnitin. In doing so, students' work will be included as source documents in the Turnitin reference database, where they will be used solely for the purpose of detecting plagiarism. Turnitin is a suite of tools that provide instructors with information about the authenticity of submitted work and facilitates the process of grading. Turnitin compares submitted files against its extensive database of content, and produces a similarity report and a similarity score for each assignment. A similarity score is the percentage of a document that is similar to content held within the database. Turnitin does not determine if an instance of plagiarism has occurred. Instead, it gives instructors the information they need to determine the authenticity of work as a part of a larger process. Please read Turnitin's Privacy Pledge, Privacy Policy, and Terms of Service, which governs users' relationship with Turnitin. Also, please note that Turnitin uses cookies and other tracking technologies; however, in its service contract with Queen's Turnitin has agreed that neither Turnitin nor its third-party partners will use data collected through cookies or other tracking technologies for marketing or advertising purposes. For further information about how you can exercise control over cookies, see Turnitin's Privacy Policy: Turnitin may provide other services that are not connected to the purpose for which Queen's University has engaged Turnitin. Your independent use of Turnitin's other services is subject solely to Turnitin's Terms of Service and Privacy Policy, and Queen's University has no liability for any independent interaction you choose to have with Turnitin.

15 Conduct

Students at Queen's must respect the Queen's University Student Code of Conduct. Also, it is expected that the following guidelines pertaining to online behaviour and equity, diversity, and inclusion be followed.

15.1 Netiquette

In the course, you may communicate with your peers and teaching team through electronic communication. You are expected to use the utmost respect in your dealings with your colleagues or when participating in activities, discussions, and online communication.

Here is a list of netiquette guidelines. Please read them carefully and use them to guide your communication in this course and beyond.

- Make a personal commitment to learn about, understand, and support your peers.
- Assume the best of others and expect the best of them.
- Acknowledge the impact of oppression on the lives of other people and make sure your writing is respectful and inclusive.
- Recognize and value the experiences, abilities, and knowledge each person brings.
- Pay close attention to what your peers write before you respond. Think through and re-read your writings before you post or send them to others.
- It is ok to disagree with ideas, but do not make personal attacks.
- Be open to being challenged or confronted on your ideas and to challenging others with the intent of facilitating growth. Do not demean or embarrass others.
- Encourage others to develop and share their ideas.

15.2 Equity, Diversity, and Inclusion (EDI)

In this course, you are expected to promote an anti-discriminatory environment where everyone feels respected, valued and welcome. It is my intent to present materials and activities that are respectful of the diversity of students and experiences in this classroom. Students in this class are encouraged to speak up and participate during class meetings. Because the class will represent a diversity of individuals, beliefs, backgrounds, and experiences, every member of this class must show respect for every other member of this class.

16 Copyright of Course Materials

Any written or visual material an instructor produces is automatically copyrighted, and an instructor may pursue any violator of that copyright whether or not a notice is placed on the course material. Copyright does not dampen any ordinary use colleagues or students would make of the material.

17 Additional Required Syllabus Information

For important information on, e.g.,

- timing of final examinations, remote exams, and remote proctoring,
- academic integrity,
- accommodations, and
- academic consideration for students in extenuating circumstances

please consult the School's common syllabus information page. The information on that page should be considered part of this syllabus.