CS 241 Foundations of Sequential Programs Winter 2024

Contents

1 Course Staff

Instructors	**Email**	**Office Hours**	**Location**
Chengnian Sun	cnsun@uwaterloo.ca	Mon 10:00–11:00am	DC 2339
		Wed 09:30–10:30am	Online
Fulltime ISA			
Hilbert Yang	cs241@uwaterloo.ca	Wed 2:30-3:30pm	MC 4065
		Thu 1:00-2:30pm	MC 4065
		Fri 10:00-11:00am	Online
Parttime ISAs			
Quan Cheng Taian	cs241@uwaterloo.ca	Tue 10:00-11:00am	MC 4065
		Thu 10:00-11:00am	MC 4065
Hassan Hashmi	cs241@uwaterloo.ca	—	—
IA			
Kris Frasheri	kfrasher@uwaterloo.ca	—	—
Instructional Support Coordinator			
Gang Lu	glu@uwaterloo.ca	Email for appointment	

In the first week of the term, there will be no office hours.

2 Lectures and Tutorials

Lectures	**Days**	**Time**	**Location**	**Instructor**
LEC 001	Tue & Thur.	11:30am–12:50pm	MC 4059	Chengnian Sun
LEC 002	Tue & Thur.	02:30pm–03:50pm	MC 4059	Chengnian Sun
Tutorials				
TUT 101	Wednesday	10:30–11:20am	MC 2035	
TUT 102	Wednesday	11:30–12:20pm	MC 2035	
TUT 103	Wednesday	01:30–02:20pm	MC 2038	

In the first week of the term, there will be no tutorials.

3 Course Description

This course presents the relationship between high-level languages and the computer architecture that underlies their implementation, including basic machine architecture, assemblers, specification and translation of programming languages, linkers and loaders, block-structured languages, parameter passing mechanisms, and comparison of programming languages.

Prerequisites: (CS 138 or 246) or (a grade of 85% or higher in one of CS 136 or 146); Computer Science and BMath (Data Science) students only. Antirequisites: CS 230, ECE 351.

Course Web Site: `http://www.student.cs.uwaterloo.ca/~cs241/` lists the course syllabus, assignment specifications and resource material.

3.1 Course Objectives

By the end of the course, students should be able to

- Write short machine- and assembly-language programs to perform simple data manipulation
- Write a basic assembler supporting labels
- Give formal specifications for regular languages, including regular expressions and bubble diagrams
- Write a scanner capable of dealing with a typical high-level programming language (given the specification)
- Give a grammar for a context-free language and create derivations for strings in a context-free language
- Write a parser for an LR(1) language given a low-level representation of the LR-parsing automaton (e.g., as derived from an automatic parser generator)
- Perform Context-Sensitive Analysis including type checking for high-level languages
- Write a simple code generator for an imperative language, i.e., one doing little or no optimization
- Apply appropriate design decisions when programming in C/C++ based on a detailed understanding of the way memory is used by a running C/C++ program

Note: When writing programs, students must be able to design, code, debug, test, and successfully run the programs.

4 Lectures and Notes

The content of the course is defined by the notes; that is, anything mentioned in lectures that is not mentioned in the notes is not a component of the course, and anything mentioned in the notes that is not mentioned in the lectures is nonetheless something students are expected to learn. The instructors will of course endeavour to cover all course content in the lectures, and to specify what they didn't cover so that students know what part of the notes to focus on. Students are advised to read the course notes regardless.

Course notes and associated videos will be made available at least a week in advance of the lecture that covers the material. Students are expected to read the course notes, but it's up to an individual student's prerogative whether they read them before or after the lectures that cover the same content. Videos are supplementary and are not a required part of the reading, but you may find them very useful for your learning.

4.1 Course Notes Availability

The course notes are available from the course web site, at `https://student.cs.uwaterloo.ca/~cs241/cgi-bin/courseNotes.php`. Only registered students may download the course notes. If you have issues accessing the course notes, contact us on Piazza.

5 Evaluation Structure

- Final exam (scheduled by Registrar's Office) worth 40%.
- Midterm exam (see schedule) worth 20%.
- 8 assignments worth 5% each for a total worth of 40%.

Additionally, you must pass the **weighted average** of the midterm exam and final exam to pass the course. If your midterm exam grade (out of 100) is M and your final exam grade (out of 100) is F, this means the following value must be 50 or higher:

$$\frac{(M \cdot 20) + (F \cdot 40)}{60}$$

Pandemic Contingency Plans: In the event that the midterm and/or final exam cannot be held in-person (e.g. due to COVID related restrictions), the weight of the cancelled exam(s) will be shifted equally across all assignments. This means that there is a chance that the assignments could end up being 100% of the grade in the event that both exams are cancelled. Additionally, if COVID restrictions require it, lectures will be cancelled. If an instructor is unable to attend their lecture because they have caught COVID or are displaying COVID symptoms, we will make an effort to find a backup instructor, but if this is not possible, that instructor's lecture will be cancelled.

6 Schedule

Week	Monday	Tuesday	Wednesday	Thursday	Friday
Week 1 Jan 8–12	**Classes begin**	Lecture 1 Coure intro & M1: Data representations		Lecture 2 M2: Machine & assembly language (part 1)	A1 out
Week 2 Jan 15–19		Lecture 3 M2: Machine & assembly language (part 2)		Lecture 4 M2: Machine & assembly language (p.3, procedures)	A2 out
Week 3 Jan 22–26		Lecture 5 M2: Assemblers M3: Regular languages (p.1)		Lecture 6 M3: Regular languages (part 2, DFAs)	A3 out **A1 due** Jan 26, 5PM
Week 4 Jan 29–Feb 2		Lecture 7 M3: Regular languages (part 3, NFAs and regexp)		Lecture 8 M3 p.4: Scanning M4 p.1: CFGs	A4 out **A2 due** Feb 2, 5PM
Week 5 Feb 5–9		Lecture 9 M4: CFGs (part 2, parse trees and parsing)		Lecture 10 M5: Top-down parsing (part 1)	A5 out **A3 due** Feb 9, 5PM
Week 6 Feb 12–16		Lecture 11 M5: Top-down parsing (part 2)		Lecture 12 M6: Bottom-up parsing (part 1, LR(0))	
Reading week					
Week 7 Feb 26–Mar 1	**A4 due** Feb 26, 5PM	Lecture 13 M6: Bottom-up parsing (part 2, SLR(1))		Lecture 14 M7: Context-sensitive analysis (part 1, types)	A6 out
Week 8 Mar 4–8		Lecture 15 M7 p.2: More types M8: Code generation (p.1)	**Midterm exam** Mar 6 4:30-6:20PM	Lecture 16 M8: Code generation (part 2)	A7–A8 out
Week 9 Mar 11–15	**A5 due** Mar 11, 5PM	Lecture 17 M8: Code generation (part 3)		Lecture 18 M8: Code generation (part 4, pointers)	
Week 10 Mar 18–22		Lecture 19 M8: Code generation (part 5, procedures)		Lecture 20 M10: Linking & loading (part 1)	**A6 due** Mar 22, 5PM
Week 11 Mar 25–29		Lecture 21 M10: Linking & loading (part 2)		Lecture 22 M8b: Optimization	**A7 due** Mar 29, 5PM
Week 12 Apr 1–5		Lecture 23 M9: Memory		Lecture 24 Summary and recap	**A8 due** Apr 5, 5PM (no slip days)
Week 13 Apr 8	**Classes end**				
Final exam (scheduling)					

The scheduling of particular lecture topics is tentative and may change without notice.

7 Assignments

There will be 8 assignments. For most students, the course material can only be learned well by carefully working through each assignment. Students are expected to compile, run, and test their own code. Feedback on the correctness of your work is provided by the Marmoset submission and testing server, https://marmoset.student.cs.uwaterloo.ca. All assignments must be submitted electronically to Marmoset. A link to the Marmoset system may also be found on the course web page.

We recommend that you start working on the assignments early. Use Marmoset to assess your progress (and grade) after convincing yourself of correctness using your own self-designed tests. By the time you submit to Marmoset, you should be convinced by your own thorough testing that your program is perfect. (Hint: the test suites and automated testing that you were introduced to in CS246 would also work very well in CS241. Consider using them here.)

Most "high-level programming" questions on assignments can be done either in C++ or Racket. However, occasionally there might be questions that restrict you to use only C++. There will also be some "low-level programming" questions that use MIPS assembly language, which you will learn about in Module 2.

There will be no hand-marking or "style marks" for assignments. Assignment marks are based entirely on passing the Marmoset tests.

7.1 Assignments by module

Module	Topics	Practiced In
1	Data Representations	Assignment 1
2	Machine & Assembly Language	Assignments 1 & 2
2	Assemblers	Assignment 3
3	Regular Languages & Automata	Assignment 4
3	Scanning	Assignment 4
4	Context-Free Grammars & Parsing	Assignment 5
5	Top-Down Parsing	Exams
6	Bottom-Up Parsing	Assignment 5
7	Context-Sensitive Analysis	Assignment 6
8	Code Generation	Assignments 7 & 8
9	Memory Management	Exams
10	Linking & Loading	Assignments 7 & 8

7.2 Assignment Slip Days/Extensions

Each student will have the ability to use up to two "slip days"—each slip day is 24 hours long—for assignments. These can be used to make late submissions on Marmoset count for marks.

If you submit late to Marmoset, this does not automatically use an extension. You must request an extension on our web form for the late submission to count:
https://student.cs.uwaterloo.ca/~cs241/cgi-bin/extension/request.cgi

Extensions can be requested *after* making your late Marmoset submission. However, you must request the extension within 72 hours of the assignment deadline.

These extensions are meant to handle cases not covered by the University's normal illness and emergency policies, ranging from simple time management problems to minor illnesses. Budget them carefully; do not waste them. If you end up using up all your assignment extensions, you will not be granted any additional extensions.

If you are using assignment extensions, you must use a whole number of them on an assignment (Example: you cannot use 1.5 assignment extensions; you must use 2 if you are 36 hours late).

Assignment extensions can be used all together on one assignment, or split across different assignments.

You do not need a reason to use assignment extensions. You may use them as you please.

Once you have made an extension request, you cannot take it back. If you make a request in error, contact the ISA at cs241@uwaterloo.ca.

If you have a standard, University-approved reason to request an extension, such as an illness with a Verified Illness Form, you do not need to use slip days. Because handling University-approved extensions is sometimes slow, you may use slip days as a "stopgap" in such situations and then request that they be refunded. If you do so, please mention it when requesting the extension.

Slip days is NOT applicable to the last assignment. That is, the due day of the last assignment cannot be extended.

7.3 Missed Assessments

You must notify the instructor of any severe, long-lasting problem that prevents you from completing an assessment. There will be no deferral, make-up, or extra credit if you miss an assessment. Under extenuating circumstances that are pre-approved within a week of the missed assessment, the instructor may shift the weight of the missed assessment to other assessments. To be considered for this option, rules instituted by the Math Faculty regarding Verification of Illness Forms will be applicable. The notification of an illness is not a guarantee that an accommodation will be made.

If a student requires an exemption from more than two assignments, they will either receive a DNW or INC depending on their performance in the rest of the course.

7.4 Marmoset downtime

If Marmoset fails to accept submissions for more than two of the six hours immediately prior to the deadline, or is down at the deadline, a 12-hour extension will be granted. For an extension to be granted, Marmoset must fail to accept submissions; failure or delay in displaying results is not grounds for extension. It is bad practice, and risky, to rely on Marmoset as your primary means of testing. The failure must be due to a problem with Marmoset, the University's network, or a widespread network failure. Your home connection is your own responsibility; if it fails, you have access to several University computer labs.

8 Submitting Assignments: Marmoset

Use Marmoset to submit and test your CS241 assignments.

There are three kinds of Marmoset tests:

- **Public tests:** These tests are "sanity checks": does your code compile and run. The results of these tests are visible as soon as the tests finish running by clicking the "view" link in the "detailed test results" column on the submission log. These tests are worth points, but there are usually few of them.

- **Release tests:** These tests are detailed tests of system behaviour. If you pass the public tests, you will be given the option to view the results of these tests. This option is on the "detailed test results" page, at the bottom. However, releasing the results costs one "release token". If you fail more than one release test, you will only be shown the first. **On most assignments, release tests are not worth points. In this case, they are to aid you in considering test cases and developing your own tests.**

- **Secret tests:** These tests are also detailed tests of system behaviour. On assignments problems in which release tests are not worth points, most secret tests are slight modifications of release tests (i.e., they are testing the same concepts as the release tests). These tests are completely hidden from you. You will not see any indication that they exist on Marmoset until three days after the assignment deadline, when they will become visible. The assignments specify how many points are secret tests; on assignments where release tests are not worth points, secret tests will be the majority.

This layout of tests is designed to encourage you to perform your own testing. For most students, it will not be possible to achieve a passing grade on the assignments without writing your own tests.

Some guidelines regarding the use of Marmoset:

a. If your submitted program does not compile or run successfully on its own, your submission will receive a result of "did not compile" and the detailed test results will contain something similar to the error message you get if you ran your program yourself. In this case, your submission will not be tested with any of the tests.

b. If your submitted program runs successfully on its own, it will be tested with all of the tests: public, release, and secret. However, you will only see the results of public tests initially.

c. If your submission fails any public test, the detailed test results will display an error message for that public test. In this case, you will not be able to view release test results for your submission until your submission passes the public tests.

d. If it passes all of the public tests, you will have the option to see information for the release tests. If you do so, you will use up one of your "release tokens" for that question to see the results of (some of) the release tests. Normally, for every assignment question, you have one release token. After using it, it will regenerate after a period of time. But you will have only a limited number of release tokens in total for each assignment question. Start your work early if you want to have more chances to see the results of the release tests. If the deadline will expire before your token regenerates, you can still submit, though you will not be able to tell how your submission did on the tests.

e. Marmoset automatically tests each submission with all of the release tests, in some order specified by the course staff. If your submission fails a release test and you use a token to see the results, you will only see the passing release tests prior to that test and the failing test in the detailed test results.

f. If you fail a release test, the information we are willing to give you for that test will be displayed by Marmoset. This may include things like the test input or the reason for the test failure. Some release tests may be *blind*, i.e., no additional information is provided. *Do not ask about or speculate about the test cases on Piazza.* The correct action when failing a release test is to re-examine your own test suite and redesign it to find the error in your code or your assumptions.

g. You can continue to submit and see the results of tests after the deadline has passed. This will not affect your mark unless you use slip days to adjust your deadline appropriately. It is a good idea to finish questions on which you ran out of time so that you can learn from the process, as most assignments build on the concepts of previous assignments.

h. Release tokens are provided as a courtesy to supplement your own testing. They are not something to which you are entitled. Release tokens can go away at any time, either as a result of Marmoset malfunctioning, or deliberately (for example, in response to widespread abuse). Loss of release tokens will not be considered grounds for assignment due date extensions.

i. Secret test details will only become visible 72 hours after the assignment deadline, to account for students who use slip days. Before this, secret tests are hidden from students completely; you will not see your score on the tests or even any indication at all that the tests exist. However, if a question has secret tests, this will be mentioned in the assignment specification.

9 Tutorials

The tutorials cover examples of the types of questions you would find on the midterm or final exam, which will aid in your understanding of the course material and help you prepare for these exams. Attendance is not mandatory but is strongly encouraged.

10 Office Hours

Some office hours are held online using MS Teams, and some are held in person. At the start of the term, make sure that you can connect to Teams. Contact CSCF if you encounter issues.

In-person office hours are on a drop-by basis; you do not need to register for them. Online office hours are by appointment managed by MS Teams.

11 Discussion Forum

CS 241 will be using Piazza to make announcements and answer questions about course material and the assignments. **You are expected to check the forum regularly,** at least once per day. Important course information will appear in pinned posts. Any information that appears in a pinned post is considered to be disseminated and we will assume that you have read it.

11.1 Rules for using Piazza

a. When asking about a particular problem on an assignment, make sure to use the appropriate folder based on the assignment number.

b. Before posting a question, read all relevant existing posts. Your question might already have been answered.

c. You may post private questions which are only visible to instructors. Note that students can show up as anonymous to other students, but not to instructors.

d. **Do not post any questions asking for hints or help with failing Marmoset release test cases.** In order to pass these test cases you should be rereading the assignment question, consulting the reference material and creating your own test cases. The instructors and staff for CS241 will never give any hints for Marmoset release test cases, and students are *strictly forbidden from doing so* as well.

12 Group Work and Collaboration

Students are required to know what constitutes academic integrity. For details, see University of Waterloo's Office of Academic Integrity website. The three most common academic offenses that CS241 students in previous terms have committed are as follows.

1. **Unauthorized collaboration:** Using a classmate's assignment as the basis or as a reference for your own or allowing someone else to do this with your assignment.

2. **Use of another student's previous assignment, test, solution:** You may not work off of, or refer to in any way, a copy of an assignment a student submitted in a previous term.

3. **Submission of another student's assignment to Marmoset:** It is a good practice to simply not share your computer with other students in the class. If you must do so, you must be extremely careful to protect your work so that you avoid anyone submitting your work and conversely, you avoid submitting someone else's work to Marmoset. By submitting to Marmoset, you are stating that the submission is your own work.

All assignments in CS241 are to be done individually. You are welcome to discuss general ideas regarding assignments with other students in the class, but no code-level sharing is permitted. You may not view someone else's code, nor share your code with someone else, either in person or via electronic communication. When code is shared, *both* parties have committed an academic offence.

Marmoset tokens cannot be shared; it is an offence to "borrow" someone else's Marmoset account for the purpose of using extra release tokens for testing, or for any other purpose.

You cannot submit a program that simply prints or returns values in order to match expected test results rather than making an actual, reasonable attempt to solve the problem as required in the assignment question specification.

If you have taken this course before, it is okay to base this term's assignments on your past assignments but you must continue to develop and refine your solution; i.e. you cannot simply submit a copy of the old assignment. We want to see that you are still spending time and effort to improve your work. It is an offence to submit for credit anything that has previously been submitted for credit in the same or any other course, unless permission is explicitly granted to do so.

Each assignment is worth a significant portion of your final grade, the penalty for an offence under Policy 71 is a grade of 0 on the assignment questions and an *additional* 5% deduction from your course grade. Additionally, be aware that if you are caught cheating on an assignment, any code from that assignment cannot be used on subsequent assignments.

12.1 Use of MOSS

MOSS (Measure of Software Similarities) is used in CS241 as a mean of comparing students' assignments in order to support academic integrity.

13 Other Policies

13.1 Intellectual Property

Students should be aware that this course contains the intellectual property of their instructor, TA, and the University of Waterloo. Intellectual property includes items such as:

- Lecture content, spoken and written (and any audio/video recording thereof);
- Lecture handouts, presentations, and other materials prepared for the course (e.g., PowerPoint slides, PDF documents);
- Questions or solution sets from various types of assessments (e.g., assignments, quizzes, tests, final exams); and
- Work protected by copyright (e.g., any work authored by the instructor or TA or used by the instructor or TA with permission of the copyright owner).

Course materials and the intellectual property contained therein, are used to enhance a student's educational experience. However, sharing this intellectual property without the intellectual property owner's permission is a violation of intellectual property rights. For this reason, it is necessary to ask the instructor, TA and/or the University of Waterloo for permission before uploading and sharing the intellectual property of others online (e.g., to an online repository).

Please alert the instructor if you become aware of intellectual property belonging to others (past or present) circulating, either through the student body or online. The intellectual property rights owner deserves to know (and may have already given their consent).

13.2 Mental Health Support

The Faculty of Math encourages students to seek out mental health support if needed.
On-campus Resources:

- Campus Wellness https://uwaterloo.ca/campus-wellness/
- Counselling Services: counselling.services@uwaterloo.ca/ 519-888-4567 ext 32655
- MATES: one-to-one peer support program offered by Federation of Students (FEDS) and Counselling Services: mates@uwaterloo.ca
- Health Services: located across the creek from the Student Life Centre, 519-888-4096.

Off-campus Resources:

- Good2Talk (24/7): Free confidential help line for post-secondary students. Phone: 1-866-925-5454
- Here 24/7: Mental Health and Crisis Service Team. Phone: 1-844-437-3247
- OK2BME: set of support services for lesbian, gay, bisexual, transgender or questioning teens in Waterloo. Phone: 519-884-0000 extension 213
- If you or anyone you know experiences any academic stress, difficult life events, or feelings like anxiety or depression, we strongly encourage you to seek support.

13.3 Diversity

It is our intent that students from all diverse backgrounds and perspectives be well served by this course, and that students' learning needs be addressed both in and out of class. We recognize the immense value of the diversity in identities, perspectives, and contributions that students bring, and the benefit it has on our educational environment. Your suggestions are encouraged and appreciated. Please let us know ways to improve the effectiveness of the course for you personally or for other students or student groups. In particular:

- We will gladly honour your request to address you by an alternate/preferred name or gender pronoun. Please advise us of this preference early in the semester so we may make appropriate changes to our records.
- We will honour your religious holidays and celebrations. Please inform of us these at the start of the course.
- We will follow AccessAbility Services guidelines and protocols on how to best support students with different learning needs.

14 University-wide Policies

Academic integrity: In order to maintain a culture of academic integrity, members of the University of Waterloo community are expected to promote honesty, trust, fairness, respect and responsibility. [Check the Office of Academic Integrity for more information.]

Grievance: A student who believes that a decision affecting some aspect of his/her university life has been unfair or unreasonable may have grounds for initiating a grievance. Read Policy 70, Student Petitions and Grievances, Section 4. When in doubt, please be certain to contact the department's administrative assistant who will provide further assistance.

Discipline: A student is expected to know what constitutes academic integrity to avoid committing an academic offence, and to take responsibility for his/her actions. [Check the Office of Academic Integrity for more information.] A student who is unsure whether an action constitutes an offence, or who needs help in learning how to avoid offences (e.g., plagiarism, cheating) or about "rules" for group work/collaboration should seek guidance from the course instructor, academic advisor, or the undergraduate associate dean. For information on categories of offences and types of penalties, students should refer to Policy 71, Student Discipline. For typical penalties, check Guidelines for the Assessment of Penalties.

Appeals: A decision made or penalty imposed under Policy 70, Student Petitions and Grievances (other than a petition) or Policy 71, Student Discipline may be appealed if there is a ground. A student who believes he/she has a ground for an appeal should refer to Policy 72, Student Appeals.

Note for students with specific learning needs: AccessAbility Services, located in Needles Hall, Room 1401, collaborates with all academic departments to arrange appropriate accommodations for students with disabilities without compromising the academic integrity of the curriculum. If you require academic accommodations to lessen the impact of your disability, please register with AccessAbility Services at the beginning of each academic term.