1 Course Staff

Instructors | Email | Office Hours
---|---|---
Sylvie Davies | sylvie.davies@uwaterloo.ca | Monday, Thursday 130pm-230pm
Nomair Naeem | nanaeem@uwaterloo.ca | Tuesday, Wednesday 12pm-1pm
ISA
Shruti Vishrant | cs241@uwaterloo.ca | Monday 3pm-5pm, Tuesday 2pm-4pm, Wednesday 1pm-3pm, Thursday 9am-11am, Friday 11am-1pm

IA
Edward Lee | e45lee@uwaterloo.ca | Friday 2pm-4pm

Instructional Support Coordinator
Gang Lu | glu@uwaterloo.ca | Email for appointment

2 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](http://www.student.cs.uwaterloo.ca/~cs241) lists the course syllabus, assignment specifications and resource material.
2.1 Course Objectives

At 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, given a grammar, produce a derivation for a given string in the 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)
• 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.

2.2 Special Note on online teaching

Since the course is being offered online, there will be no lectures. Instead, course content will be made available in text format along with supplemental videos for illustrating specific topics, as needed. The course material will be made available well ahead of assignment deadlines so that students can organize their term to fit their needs.

<table>
<thead>
<tr>
<th>Module</th>
<th>Topics</th>
<th>Practiced In</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Number Representations</td>
<td>Assignment 1</td>
</tr>
<tr>
<td>2</td>
<td>Machine Language</td>
<td>Assignment 1</td>
</tr>
<tr>
<td>2</td>
<td>Assembly Language</td>
<td>Assignment 2</td>
</tr>
<tr>
<td>2</td>
<td>Assemblers</td>
<td>Assignment 3</td>
</tr>
<tr>
<td>3</td>
<td>Regular Languages</td>
<td>Assignment 4</td>
</tr>
<tr>
<td>3</td>
<td>Scanning</td>
<td>Assignment 5</td>
</tr>
<tr>
<td>4</td>
<td>Context-free Grammars</td>
<td>Assignment 5 &amp; 6</td>
</tr>
<tr>
<td>5</td>
<td>Top Down Parsing</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Bottom Up Parsing</td>
<td>Assignment 6</td>
</tr>
<tr>
<td>7</td>
<td>Context-Sensitive Analysis</td>
<td>Assignment 7</td>
</tr>
<tr>
<td>8</td>
<td>Code Generation</td>
<td>Assignment 8 &amp; 9</td>
</tr>
<tr>
<td>9</td>
<td>Compiler Optimizations</td>
<td>Assignment 9</td>
</tr>
<tr>
<td>10</td>
<td>Memory Management</td>
<td>Assignment 10</td>
</tr>
<tr>
<td>11</td>
<td>Linking and Loading</td>
<td>Assignment 10</td>
</tr>
</tbody>
</table>
## 3 Printable Schedule

<table>
<thead>
<tr>
<th>Week</th>
<th>Dates</th>
<th>Complete Reading by:</th>
<th>Complete Quiz by:</th>
<th>Assignment due by:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>May 10th - 14th</td>
<td>Module 1: till end</td>
<td>Quiz On Module 1:</td>
<td>Assignment 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Module 2: till (including) Section 3.5</td>
<td>Wed, May 19th, 5pm</td>
<td>Fri, May 21st, 5pm</td>
</tr>
<tr>
<td>2</td>
<td>May 17th - 21st</td>
<td>Module 2: till end</td>
<td>Quiz On Module 1:</td>
<td>Assignment 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Quiz On Module 2:</td>
<td>Fri, May 28th, 5pm</td>
</tr>
<tr>
<td>3</td>
<td>May 24th - 28th</td>
<td>Module 3: till (including) Deterministic Finite Automata</td>
<td>Quiz On Module 2:</td>
<td>Assignment 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Wed, May 26th, 5pm</td>
<td>Fri, June 25th, 5pm</td>
</tr>
<tr>
<td>4</td>
<td>May 31st - June 4th</td>
<td>Module 3: till end</td>
<td>Quiz On Module 3:</td>
<td>Assignment 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Quiz On Module 3:</td>
<td>Fri, June 11th, 5pm</td>
</tr>
<tr>
<td>5</td>
<td>June 7th - 11th</td>
<td>Module 4: till end</td>
<td>Quiz On Module 4:</td>
<td>Assignment 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Quiz On Module 4:</td>
<td>Fri, June 18th, 5pm</td>
</tr>
<tr>
<td>6</td>
<td>June 14th - 18th</td>
<td>Module 5: till end</td>
<td>Quiz On Module 5:</td>
<td>Assignment 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Quiz On Module 5:</td>
<td>Fri, June 25th, 5pm</td>
</tr>
<tr>
<td>7</td>
<td>June 21st - 25th</td>
<td>Module 6: initial read</td>
<td>Quiz On Module 6:</td>
<td>Assignment 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Wed, June 23rd, 5pm</td>
<td>Fri, July 9th, 5pm</td>
</tr>
<tr>
<td>8</td>
<td>June 28th - July 2nd</td>
<td>Module 6: till end</td>
<td>Quiz On Module 6:</td>
<td>Assignment 7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Quiz On Module 6:</td>
<td>Fri, July 16th, 5pm</td>
</tr>
<tr>
<td>9</td>
<td>July 5th - 9th</td>
<td>Module 7: till end</td>
<td>Quiz On Module 7:</td>
<td>Assignment 8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Quiz On Module 7:</td>
<td>Fri, July 23rd, 5pm</td>
</tr>
<tr>
<td>10</td>
<td>July 12th - 16th</td>
<td>Module 8: till (including) Control-flow Statements</td>
<td>Quiz On Module 7:</td>
<td>Assignment 9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Quiz On Module 7:</td>
<td>Thurs. Aug 5th 11:59pm</td>
</tr>
<tr>
<td>11</td>
<td>July 19th - 23rd</td>
<td>Module 8: till end</td>
<td>Quiz On Module 8:</td>
<td>Assignment 10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Module 9: till end</td>
<td>Quiz On remaining Module 8 and 9:</td>
<td>Fri. Aug 13th 11:59pm</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Quiz On remaining Module 8 and 9:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Quiz On remaining Module 8 and 9:</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>July 26th - 30th</td>
<td>Module 10: till end</td>
<td>Quiz On Module 10 and 11:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Module 11: initial read</td>
<td>Quiz On Module 10 and 11:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Quiz On Module 10 and 11:</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Aug 2nd - Aug 5th</td>
<td>Module 11: till end</td>
<td>Quiz On Module 10 and 11:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Quiz On Module 10 and 11:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Quiz On Module 10 and 11:</td>
<td></td>
</tr>
<tr>
<td>Exam</td>
<td>Aug 7th - Apr 16th</td>
<td></td>
<td>Quiz On Module 10 and 11:</td>
<td>Assignment 10</td>
</tr>
<tr>
<td>Period</td>
<td></td>
<td></td>
<td>Quiz On Module 10 and 11:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Quiz On Module 10 and 11:</td>
<td></td>
</tr>
</tbody>
</table>
4 Evaluation Structure

The course grade will be based off 10 assignments (each worth 9%) and 10 Quizzes (each worth 1%).

\[
\text{Grade} = (\text{Quiz\%} \times 10\%) + (\text{Assignments\%} \times 90\%)
\]

Typically, no assignments can be due beyond the last day of classes. The Math faculty has allowed one assessment during the final exam period. Therefore, we have replaced the standard final exam with assignment 10.

This term we are piloting a new (and free) platform named Perusall to encourage collaborative reading of course notes. You can earn up to 3% in bonus marks by using this platform. These bonus marks will be assigned based on the quality of annotations and other engagement metrics such as active participation on the platform. Since this is the first term we are experimenting with the platform we are unable to clearly pinpoint how much effort will be required to get the full bonus. For a little more detail, see Section 5.

4.1 Assignments

There will be ten assignments. For most students, the course material can only be learned well by carefully working through each assignment. Real-time 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 and results are normally quickly available.

We recommend that you start working on the assignments early. Use Marmoset to assess your progress (and grade) afterconvincing 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 CS 246 would also work very well in CS 241. Consider using them here.) A link to the Marmoset system and instructions for using it may also be found on the course web page.

Most programming questions on assignments can be done in C++/Racket. However, occasionally there might be questions that restrict you to use only C++.

4.2 Assignment Slip Days/Extensions

Each student will have the ability to use up to three 24 hour extensions for assignments (not quizzes). These extensions are meant to handle emergencies close to assignment deadlines or short 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.

Extensions can be used after the assignment deadline, and will apply retroactively. However, they must be used within 72 hours of the original assignment deadline.
You do not need a reason to use assignment extensions, you may use them as you please.

You cannot use an assignment extension on A10 (due during Exam period). This is to ensure that course staff has appropriate time to compile the course grades and submit them to the Registrar’s Office.

Once you have made an extension request, you cannot take it back.

Details on how to request an extension will be posted on Piazza.

### 4.3 Missed Assessments

You must notify the instructor of any severe, long-lasting problem that prevents you from completing an assessment. There will be no deferred, makeup or extra credit for a missed assessment. Under extenuating circumstances, that are pre-approved within a week of the missed assessment, the instructor may assign a higher weight to later assessments. To be considered for this option, any rule instituted by the Math Faculty regarding the Verification of Illness, will be applicable. The notification of an illness is not a guarantee that an accommodation will be made. If a student requests 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.

### 4.4 Remark Request for Assessments

Requests for regrades for hand-marked assignments will be accepted up to five (5) days after students have the opportunity to view their assessment. Details of how to request a regrade will be posted in Piazza.

### 4.5 Hand-Marking

From time to time, we may choose to hand-mark an assignment question, in addition to the regular marking performed by Marmoset. The purpose of hand-marking is to review your submission and help you write better code. Although we may assign a few marks to hand-marking, its real purpose is to give you specific ways to help you improve the quality of your code. To encourage you to always write good code, we will not announce ahead of time which assignments will be hand-marked.

### 5 Perusall

The course content is taught primarily through course notes that are available as PDF documents. It is therefore important that you spend time to actively read and understand these notes. Recently, the course instructors became aware of the Perusall Platform which markets itself as a social e-learning experience. The idea is that instead of students reading the notes in their own silos, they can read online and pose questions or make observations from within the platform. Other students can answer any questions or discuss observations. Occasionally, the course staff might also jump in such as when students are unable to come to a consensus.

In the past few terms (online teaching), students have sometimes made use of one-on-one office hours and most often used Piazza to get answers to their questions regarding course notes. We think that being able to pose questions from right within the text, while you read it, will be convenient. While reading the text, you will see questions/observations made by others which might quickly answer your questions or give you other things to think about. This should make for a nicer and more convenient learning space compared to having to go to a secondary source (Piazza) to see what others have struggled with. The success of this
platform is entirely dependent on how you, as the students, engage with the material. We will continue to use Piazza for all other non-course notes related questions.

6 Office Hours

All office hours will be held remotely using MS Teams. At the start of the term, make sure that you can connect to Teams. Contact CSCF if you encounter issues. Office hour appointments will be booked online. Visit the following website to get details: Office Hours Page

7 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.

7.1 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 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.

8 Submitting Assignments: Marmoset

Use Marmoset to submit and test your CS241 assignments.

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 public tests.

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 release test your submission until your submission passes the public test.
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. Normally, for every assignment question, you will be initially given 3 release tokens. If you use up one or more of them, one release token will regenerate once every 12 hours, until you have 3 release tokens again. 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 that test and one more test in the detailed test results. If your submission passes all the release tests, you will not see any release tests in the detailed test results, but you will be credited with full marks for that question.

f. If you fail a release test, the information we are willing to give you for that test will be displayed by Marmoset. You will not be given additional details. Some tests are blind, i.e., no additional information is provided. Again, 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 result of release tests after the deadline has passed. It’s a good idea to finish questions on which you ran out of time, to make sure that you’ve done all the learning.

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. Some questions might have secret tests. Secret test details (whether your submission passed or failed the test) will only become visible a few days after the assignment deadline (this process is manually activated by the course staff).

8.1 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 or a widespread network failure. Your home connection is your own responsibility.

9 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. Excessive 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 Marmose, 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.

Although each assignment is worth a small portion of your final grade, the penalty for an offence under Policy 71 is a grade of 0 on the assignment and an additional 5% deduction from your course grade.

9.1 **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.

10 **Other Policies**

10.1 **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).

10.2 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.

10.3 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.

11 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.