

CS 241 Foundations of Sequential Programs Spring 2018

Instructors:

Brad Lushman, brad.lushman@uwaterloo.ca, DC 3110, Wednesdays 11:30–1:00pm
Nomair Naeem, nomair.naeem@uwaterloo.ca, DC 3111, Wednesdays 11:00–12:30pm
Kevin Lanctot, kevin.lanctot@uwaterloo.ca, DC 2131, Tuesdays 12:00–1:30pm

ISA:

Ramin Zarifi, cs241@uwaterloo.ca

Senior ISA:

Sean Harrap sharrap@uwaterloo.ca

Instructional Support Coordinator:

Gang Lu (glu@uwaterloo.ca, MC 4008, x38243)

Lecture Time	Location	Lecturer
10:00 – 11:20am, TTh	PHY 313	Brad Lushman
1:00 – 2:20pm, TTh	MC 4061	Nomair Naeem
2:30 – 3:50pm, TTh	MC 4061	Nomair Naeem
2:30 – 3:50pm, TTh	RCH 305	Kevin Lanctot

Objective: To describe the relationship between high-level programming languages and the computer architecture that underlies their implementation.

Prerequisites: CS 246 or (CS 138 and enrolled in Software Engineering); Computer Science students only.

Antirequisites: CS 230, GENE 344

Website: <http://www.student.cs.uwaterloo.ca/~cs241>

Essential announcements and assignment specifications can be found here.

1 Discussion Forum

CS 241 will be using Piazza: <https://piazza.com/>

We encourage you to use the forum. It is a handy and timely way for students to learn from each other and the course staff. Essential announcements will also be posted here. **You are expected to check the forum regularly, at least once per day. Any information that appears in the forum is considered to be disseminated, and we will assume that you have read it.**

1.1 Rules for using Piazza

Make sure to follow the following rules of using the forum:

- When asking about a particular problem on an assignment put the following in your title, AxxxPyyy. E.g. if you asking about Problem 2 on Assignment 1 include A1P2 in the title.
- Before posting a question, read all relevant existing posts. Your question might already have been answered. **Instructors will ignore questions that have already been answered.**
- 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.

- **You may NOT post any questions asking for hints/help with failing Marmoset test cases.** The forum is NOT intended to help you pass Marmoset tests. The purpose of assignments is to give you exercise in the concepts and skills discussed in lectures. The point is not to earn marks by any means possible; marks are a consequence of the understanding gained through practice. Any such questions will be immediately deleted. Accounts of repeat offenders will be suspended. The instructors and staff for CS241 will NEVER give any hints for Marmoset BLIND test cases, and students are STRICTLY FORBIDDEN from doing so as well.

2 Marmoset

CS241 uses Marmoset to test all student assignments.

- 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.
- If your submitted program runs successfully on its own, it will be tested with all of the public tests.
- If it fails any public test, the detailed test results will display an error message for that public test. In this case, your submission will not be tested with any of the release tests.
- 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.
- 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.
- If you fail a release test, you may get a very small amount of information about what went wrong. You will not be given details of the test case that you failed. **Do not attempt to guess what that test case might be; do not ask about it on Piazza, and do not speculate about 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. The purpose of assignments is to give you exercise in the concepts and skills discussed in lecture. The point is not to earn marks by any means possible; marks are a consequence of the understanding gained through practice.
- You can continue to submit and see the result of release tests after the deadline has passed right up until 11:55 pm on the last day of lectures.
- **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.

2.1 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.

3 Hand-Marking, Code Reviews

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 (largely so that it is taken seriously), its real purpose is to hopefully give you specific ways to help you improve the quality of your code. We may also, on occasion, solicit volunteers to have their code reviewed live during tutorials.

4 Tutorials

In the first week of the term, there will be no tutorial. Regular tutorials will begin from week two and will be held in the scheduled room listed below.

<u>Section</u>	<u>Time</u>	<u>Location</u>
101	12:30-1:20W	MC 4042
102	9:30-10:20W	MC 4042
103	1:30-2:20W	MC 4042
104	10:30-11:20W	MC 4042
105	8:30-9:20W	MC 4042

5 Course Description

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.

6 Course Overview

Machine architecture and assembly language (6 hours)

Functional components of a computer: memory, control unit, arithmetic/logic unit, input/output devices. Data representation. Machine language: operation codes, addressing modes, indexing, base registers, register designation.

Assemblers, linkers, and loaders (6 hours)

Mnemonic op-codes, pseudo-ops, symbolic constants and addresses, literals. Assembler algorithm, linker and loader algorithms

Regular languages and scanning (5 hours)

Architecture of a compiler. Syntax vs. semantics. Introduction to formal languages. Regular languages, regular expressions and finite state machines.

Context-free languages and parsing (8 hours)

Context-free grammars, derivations, derivation trees, ambiguous grammars. Introduction to top-down and bottom-up parsing, LL(1) and LR(1) grammars. Tool-based parser generation.

Semantic Analysis and Code generation (6 hours)

Constructing parse trees. Type-checking and scope. Code generation.

Memory Management/Topics of Interest (5 hours)

Implications of stack versus heap allocation. Dynamic Memory Allocation. Automatic Garbage Collection. Other topics to be introduced at the instructor's discretion.

7 Assignments

There will be eleven assignments. For most students, the course material can only be learned well by carefully working through each and every assignment. Real-time feedback on the correctness of your work is provided by the Marmoset submission and testing server. All assignments must be submitted electronically to Marmoset and results are normally quickly available. We recommend that you work on exercises early and often. 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 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 be found on the course web page.

The Marmoset system will accept submissions until 11:55pm on the last day of classes. However, to receive full credit, a submission *must* be received by on time on the due date. There will be a late penalty of -25% any additional marks gained for submissions that are up to 24 hours late. After 24 hours, the late penalty will be -50% for any marks gained until 11:55 pm on the last day of lectures (July 25). Your current mark for an assignment will never decrease as a result of submitting late. We will take the maximum of your on time submission, your submissions up to 24 hours late, and you submission up until the last day.

You must notify the instructor of any severe, long-lasting problem that prevents you from completing an assignment and submit the current version of the Verification of Illness form, <https://uwaterloo.ca/campus-wellness/sites/ca.campus-wellness/files/uploads/files/VIF-online.pdf>.

8 Exams

Midterm Exam: The midterm exam is on Wednesday, June 20, 2018 from 7:30pm to 9:20pm. Rooms will be announced later.

Final Exam: There will be a $2\frac{1}{2}$ -hour final exam held during the examination period.

8.1 No Makeup Arrangements for Midterm

There will be no deferred/makeup midterm exam. Under extenuating circumstances that are pre-approved, where a student is unable to write the mid-term, the instructor will assign a higher weight to students final exam. To be considered for this option you must submit the current version of the Verification of Illness form.

8.2 Regrading Request for Midterm

Requests for regrading will be accepted up to 14 days after students have the opportunity to pick up their midterm. Details of how to request a regrade will be posted in Piazza after the midterm has been handed back.

9 Grades

Normally Computed Grade = (Final% \times 50%) + (Midterm% \times 25%) + (Assignments% \times 25%)

$$\text{Weighted Exam} = (\text{Midterm}\% + 2 \times \text{Final}\%) / 3$$

The Weighted Exam mark should be 50% or more in order to pass the course. If not, then the final course grade is the lesser of (normally computed grade, WeightedExam mark).

10 Academic Offenses

Students are expected to know what constitutes academic integrity, to avoid committing academic offenses, and to take responsibility for their actions. Students who are unsure whether an action constitutes an offense, or who need help in learning how to avoid offenses (e.g., plagiarism, cheating) or about rules for group work/collaboration should seek guidance from the course professor, ISC, ISA, TA, IA, academic advisor, or the Undergraduate Associate Dean.

The two 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 students 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.

See <https://uwaterloo.ca/academic-integrity/integrity-students> for details.

For information on categories of offenses and types of penalties, students should refer to Policy #71, Student Academic Discipline, <http://www.adm.uwaterloo.ca/infosec/Policies/policy71.htm>. Students who believe that they have been wrongfully or unjustly penalized have the right to grieve; refer to Policy #70, Student Grievance, <http://www.adm.uwaterloo.ca/infosec/Policies/policy70.htm>.

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 look at someone else's code, nor share your code with someone else, either in person or via electronic communication (e.g., instant messaging, course newsgroup). When code is shared, **both** parties are considered to have committed an 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.

MOSS (Measure of Software Similarities) is used in this course as a mean of comparing students' assignments.

If you have taken this course before, we require that you do each assignment from scratch. 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 only about $2\frac{1}{4}\%$ 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.

11 Academic Accommodations

AccessAbility Services (AAS), located in Needles Hall 1132, 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 the AAS at the beginning of each academic term.

12 Intellectual Property

Students should be aware that this course contains the intellectual property of their instructor, TA, and/or 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);
- 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 Course Schedule

A course outline follows. Note that the assignment of topics to weeks is our current best guess as to what will be covered when. If you miss a lecture, it is your responsibility to get the material you missed from a friend or nearby classmate. **Assignment due dates are subject to change as dictated by lecture pace.**

Week	Lecture Dates	Topics	Notes
I	May 1, 3	Introduction / Machine Language	No assignment due
II	May 8, 10	Machine Language	No assignment due
III	May 15, 17	Assembly Language	Assignment 1 due Mon, May 14
IV	May 22, 24	Assemblers	Assignment 2 due Tues, May 22
V	May 29, 31	Regular Languages (RL)	Assignment 3 due Wed, May 30
VI	June 5, 7	RL/ Context-free Grammars	Assignment 4 due Wed, June 6
VII	June 12, 14	Context-free Grammars	Assignment 5 due Fri, June 15
VIII	June 19, 21	CFGs & Parsing	<i>Midterm: June 20, 7:30-9:20</i>
IX	June 26, 28	Context-Sensitive Analysis	Assignment 6 due Wed, June 27
X	July 3, 5	Code Generation, Optimization	Assignment 7 due Wed, July 4
XI	July 10, 12	Memory Management	Assignment 8 due Wed, July 11
XII	July 17, 19	Linking and Loading	Assignment 9 due Wed, July 18
	July 25	Last day of classes	Assignment 10 due Wed, July 25