CS 241 Foundations of Sequential Programs Fall 2017

1 Course Staff

Instructors	Email	Office	Office Hours
Dan Holtby	djholtby@uwaterloo.ca	DC 2130	Fri. 10:30 –11:30 am
Sean Kauffman	sean.kauffman@uwaterloo.ca	E5 4111	Wed. 9:30 –10:30 am
Kevin Lanctot	kevin.lanctot@uwaterloo.ca	DC 2131	Wed. 2:15 – 4:15 pm
ISA			
Bing Xu Hu	cs241@uwaterloo.ca	MC 4065	Wed. 5:30pm – 6:30 pm Thurs. 4:00pm– 5:00 pm
Part-time ISA			
Pierre Guidez	pierre-louis.guidez@uwaterloo.ca	MC 4065	email for appt.
Instructional Support Coordinator			
Gang Lu	glu@uwaterloo.ca	MC 4008	email for appt.

2 Lectures and Tutorials

Lectures	Days	Time	Location	Lecturer
LEC 001	Tue. & Thur.	8:30 – 9:50am	STC 50	Sean Kauffman
LEC 002	Tue. & Thur.	11:30 – 12:50pm	MC 1056	Kevin Lanctot
LEC 003	Tue. & Thur.	1:00 – 2:20pm	MC 1056	Kevin Lanctot
LEC 004	Tue. & Thur.	2:30 – 3:50pm	RCH 305	Dan Holtby
Tutorials				
TUT 101	Wednesday	2:30–3:20	MC 1056	Bing Xu Hu
TUT 102	Wednesday	12:30–1:20	RCH 207	Pierre Guidez
TUT 103	Wednesday	3:30–4:20	MC 4060	Bing Xu Hu
TUT 104	Wednesday	4:30–5:20	STC 0020	Pierre Guidez
TUT 105	Wednesday	4:30–5:20	MC 4058	Bing Xu Hu

3 Course Overview

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 students only. Antirequisites: CS 230.

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

3.1 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.

3.2 Topics Covered

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

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

Context-free languages and parsing
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
Constructing parse trees. Type-checking and scope. Code generation.

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

3.3 Course Schedule

The assignment of topics to weeks may vary slightly from this schedule. If you miss a lecture, it is your responsibility to get the material you missed from a friend or nearby classmate.

Week	Lecture Dates	Topics	Notes
I	Sept. 7, 12	Introduction / Machine Language	No assignment due
II	Sept. 14, 19	Machine Language	Assignment 1 due Fri, Sept. 22
III	Sept. 21, 26	Assembly Language	Assignment 2 due Fri, Sept. 29
IV	Sept. 28, Oct 3	Assemblers	Assignment 3 due Fri, Oct 6
V	Oct. 5, 10	Regular Languages (RL)	Assignment 4 due Fri, Oct. 13
VI	Oct. 12, 19	RL/ Context-free Grammars	Assignment 5 due Fri, Oct. 20
VII	Oct. 24, 26	Context-free Grammars	*Midterm: Wed, Oct. 25, 7:00-8:20*
VIII	Oct. 31, Nov. 2	Parsing	Assignment 6 due Fri, Nov. 3
IX	Nov. 7, 9	Context-Sensitive Analysis	Assignment 7 due Fri, Nov. 10
X	Nov. 14, 16	Code Generation, Optimization	Assignment 8 due Fri, Nov. 17
XI	Nov 21, 23	Memory Management	Assignment 9 due Fri, Nov. 24
XII	Nov. 28, 30	Linking and Loading	Assignment 10 due Fri, Dec. 1

4 Evaluation Structure

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

Weighted Exam = (Midterm% + 2 × 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).

4.1 Assignments

There will be ten 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. 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 credit, a submission *must* be received before 5pm on the due date.

4.2 Hand-Marking and 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.

4.3 Late and Missed Assignments

There will be a late penalty of -25% for assignment submitted up to 24 hours late. After 24 hours, late assignments will receive 0.

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. The weight of your other assignments will be increased to cover the missed assignment.

4.4 Exams

Midterm Exam: The midterm exam is on Wednesday, October 25, 2017 from 7:00pm to 8:20pm (not until 8:50 as stated in Quest). Rooms will be announced later.

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

4.5 No Makeup Arrangements for Midterm or Final

There will be no deferred or makeup for the midterm or final exam. Under extenuating circumstances that are pre-approved, where a student is unable to write the midterm, the instructor will assign a higher weight to the final exam. To be considered for this option you must submit the current version of the Verification of Illness form. If a student misses the final exam, with a valid, documented reason, they will either receive a DNW or INC depending on their performance in the rest of the course.

4.6 Regrading Request for Midterm

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

5 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.

5.1 Rules for using Piazza

a. When asking about a particular problem on an assignment put the following in your title, AxxPyy. E.g. if you asking about Problem 2 on Assignment 1 include A1P2 in the title.

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. **You may NOT post any questions asking for hints or help with failing Marmoset 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.

6 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 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.

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, 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 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, though post-deadline submissions do not affect marks. 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.

6.1 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.

7 Group Work and the Limits of Acceptable Collaboration

Students are required to know what constitutes academic integrity. See https://uwaterloo.ca/academic-integrity/integrity-students for details. 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.

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.

If you have taken this course before, we may 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 2.5% 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.

8 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);
- 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).

9 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 departments 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 disabilities: 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.