CS 241 Foundations of Sequential Programs Fall 2016

Instructors: Troy Vasiga (`troy.vasiga AT uwaterloo DOT ca`)
Matt Crane (`matt.crane AT uwaterloo DOT ca`)
ISA: Max Bardakov (`cs241@uwaterloo.ca`)
Instructional Support Coordinator: Gang Lu (`glu@uwaterloo.ca`, MC 4008, x38243)

Lecture Time	**Location**
2:30 – 3:50pm TTh	RCH 309
10:00 – 11:20am TTh	STC 0020
8:30 – 9:50am TTh	STC 0020
11:30am – 12:50pm TTh	MC 2035

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

Prerequisites: CS 246; 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:

- 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, 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.
- **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.

Section	Time	Location
101	3:30-4:20W	MC 2035
102	1:30-2:20W	MC 2038
103	2:30-3:20W	MC 2035
104	4:30-5:20W	MC 1056
105	10:30-11:20W	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 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 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 credit, a submission *must* be received before 7pm on the due date (except that the last assignment will be due at 11:55pm).

8 Exams

Midterm Exam: There will be one 1-hour 50 minutes midterm exam on Wednesday, October 26th, 2016 from 7:00pm to 8:50pm. Rooms will be announced later.
Final Exam: There will be a $2\frac{1}{2}$-hour final exam held during the examination period.

9 Grades

Final Exam: 50%; Midterm Exam: 25%; Assignments: 25%

WeightedExam = (Midterm% + 2 * Final%) / 3

The WeightedExam mark should be 50% or more in order to pass the course. If the WeightedExam mark is less than 50%, then the final course grade is the lesser of (your normally computed grade, your 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. 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. **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% 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.

Note for students with disabilities: 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.

11 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	Sep 8	Introduction / Machine Language	No assignment due
II	Sep 13, 15	Machine Language	No assignment due
III	Sep 20, 22	Assembly Language	Assignment 1 due Th Sep 22
IV	Sep 27, 29	Assemblers	Assignment 2 due Th Sep 29
V	Oct 4, 6	Linking and Loading	Assignment 3 due Th Oct 6
VI	Oct 13	Reg. Lang. (no class Tues, Reading Day)	Assignment 4 due Th Oct 13
VII	Oct 18, 20	RL/ Context-free Grammars	Assignment 5 due Th Oct 20
VIII	Oct 25, 27	CFGs / Parsing	Midterm: Oct 26, 7:00pm to 8:50pm
IX	Nov 1, 3	Parsing	Assignment 6 due Th Nov 3
X	Nov 8, 10	Context-Sensitive Analysis	Assignment 7 due Th Nov 10
XI	Nov 15, 17	Code Generation, Optimization	Assignment 8 due Th Nov 17
XII	Nov 22, 24	Memory Management	Assignment 9 due Th Nov 24
XIII	Nov 29, 31	Additional topics	No assignment due
XIV	—	No lecture (classes end M Dec 5)	Assignment 10 due M Dec 5