CSci 2041: Advanced Programming (Evening)

University of Minnesota
Department of Computer Science and Engineering
Fall 2016

Last Revision September 6, 2016

Principles and techniques for creating correct, robust, and modular programs. Computing with symbolic data, recursion and induction, functional programming, impact of evaluation strategies, parallelism. Organizing data and computations around types. Search-based programming, concurrency, modularity. Practical application. Weekly labs.

Lecture:	10	**Keller Hall** 3–111	6:30 PM–9:00 PM	Wednesday
Labs:	11	**Keller Hall** 1–250	3:35 PM–4:25 PM	Tuesday
	12	**Keller Hall** 1–250	4:40 PM–5:30 PM	Tuesday

I also teach another version of CSci 2041 that meets Monday, Wednesday, and Friday mornings. It is equivalent to the evening version. However, the two versions have different syllabi, different tests, and may have different assignments.

Instructor.

I'm James B. Moen. Since this course has a large enrollment, teaching responsibilities are divided between myself, the graduate TA's, and the undergraduate TA's. I teach the lecture section of the course; I write lab assignments, projects, and tests; I answer students' questions during my office hours and by email.

Name:	James B. Moen
Email:	`moen0017@umn.edu`
Office:	**Shepherd Laboratories** 583
Office Phone:	(612) 625–7905
Office Hours:	1:00 PM–2:00 PM Monday, Wednesday, Friday (or by appointment)

I may be in my office at other times, not just during my scheduled office hours. You may visit my office (almost) any time I am there. You may also telephone or send email at any time. However, if you telephone and I am not in the office, then please leave me a message by email, not voice mail. I have many email addresses; if possible, please send email about CSci 2041 to me at the address shown above.

Teaching assistants.

Samuel Marquart and Ancy Tom are the graduate TA's for this course. Isaac Cortes Mera, Charles Harper, Lucas Meyers, and Shannyn Telander are the undergraduate TA's. The graduate TA's supervise the undergraduate TA's; they answer students' questions during their office hours, and by email. The undergraduate TA's run the labs; they grade lab assignments, projects, and tests; they answer students' questions during their office hours, and by email.

NAME	EMAIL	OFFICE	OFFICE HOURS	LABS
Samuel Marquart (GRAD)	`marqu317@umn.edu`	To be announced	To be announced	
Ancy Tom (GRAD)	`tomxx030@umn.edu`	To be announced	To be announced	
Isaac Cortes Mera	`corte114@umn.edu`	To be announced.	To be announced.	11, 12
Charles Harper	`harpe334@umn.edu`	To be announced	To be announced	11, 12
Lucas Meyers	`meye2058@umn.edu`	To be announced	To be announced	12
Shannyn Telander	`telan034@umn.edu`	To be announced	To be announced	11

Shannyn Telander	telan024@umn.edu	To be announced	To be announced	11

Note that *the TA's do all the grading for this course!* If you have questions about your grades, on a lab assignment, a project, or a test, then *please contact one of the TA's first.* If you contact me first, then I will refer you to the TA's.

Textbooks.

There are no required textbooks for this course: everything you need to know will be discussed in the lectures. As a result, it is important that you not miss any lectures. If you do, then you may fall behind and have difficulty with course work. Please plan accordingly.

Why are there no required textbooks? (1) We will be using the programming language Clojure. Since Clojure is relatively new, there are few good textbooks written for it yet. (2) Many topics discussed in this course are not traditionally part of the undergraduate curriculum. Although there is much published work about these topics, most of it is not in textbooks.

You may find the **Clojure Web Site** to be a useful source of information. There is also much additional information about Clojure available online, although it varies in quality.

Grading.

This course will include fourteen weekly laboratory assignments, three programming projects, two midterm examinations, and a final examination. Your final grade will be determined by the number of points you receive on each. 10% of your grade will be based on the lab assignments, 30% on the programming projects, 30% on the midterm examinations, and 30% on the final examination. Here is how your final grade will be computed. Let l, L, p, P, m, M, f, and F be defined like this:

l = Points you received on all labs
L = Points possible on all labs
p = Points you received on all projects
P = Points possible on all projects
m = Points you received on both midterm examinations
M = Points possible on both midterm examinations
f = Points you received on the final examination
F = Points possible on the final examination

A real number G will be calculated according to the following formula, where $0 \leq G \leq 1$.

$$G = 0.1 \frac{l}{L} + 0.3 \frac{p}{P} + 0.3 \frac{m}{M} + 0.3 \frac{f}{F}$$

G will then be converted to an integer percent, rounding up to the next larger integer in the usual way. This table shows what your final grade will be, depending on that percent. For S/N grading, an S requires that you have at least 60%. The performance of other students will not affect your grade.

PERCENT G	GRADE
$95\% \leq p \leq 100\%$	A
$90\% \leq p \leq 94\%$	A−
$85\% \leq p \leq 89\%$	B+
$80\% \leq p \leq 84\%$	B
$75\% \leq p \leq 79\%$	B−
$70\% \leq p \leq 74\%$	C+
$65\% \leq p \leq 69\%$	C

$65\% \le p \le 69\%$	C
$60\% \le p \le 64\%$	C−
$55\% \le p \le 59\%$	D+
$50\% \le p \le 54\%$	D
$0\% \le p \le 49\%$	F

Note that *the percent shown on the course Moodle site is not G!* Moodle's percent is a simple percentage of the total points you receive. Do not confuse Moodle's percent with *G;* it is not used to compute your grade.

If you believe that a laboratory assignment, a programming project, or an examination has been graded incorrectly, then you must contact the TA's within *ten working days* after the grade is posted on Moodle. If you believe that your final course grade is incorrect, then you must contact me within *ten working days* after the grade appears on the University's web site. In either case, after ten working days, I will not change your grade.

Laboratories.

Most weeks, there will be a one-hour computing lab. You will work with a partner during each lab, and both partners will receive the same number of points for their work. With the lab instructor's permission, you may work alone instead, but I recommend that you work with a partner. You need not work with the same partner each time. Although you may discuss the labs with others in a general way, you are not allowed to work on a lab with anyone other than your partner.

Late labs may be turned in up to 24 hours late, but with a 10% penalty. If your lab is more than 24 hours late, then it will not be accepted.

Here is how to make up a missed computing lab. (1) Get the lab assignment from Moodle. (2) Do the lab assignment on your own. (3) Turn in the lab assignment before it is due. If you don't know how to turn in the lab assignment, then ask your lab TA. If you must turn in a lab assignment late without penalty (perhaps because of illness), then ask your lab TA for permission.

Projects.

There will be three programming projects, all in Clojure. Projects must be done individually, although you may discuss them with others in a general way.

Late projects may be turned in up to 24 hours late, but with a 10% penalty. If your project is more than 24 hours late, then it will not be accepted. If you don't know how to turn in a project, then ask your lab TA. If you must turn in a project late without penalty (perhaps because of illness), then ask your lab TA for permission.

Examinations.

There will be two midterm examinations, and a final examination. All three must be done individually. They will be "open notes," so you will be allowed to use anything represented on paper to help you answer questions, including class notes, lab assignments, program listings, and Xeroxed pages (this may not be a complete list). However, you will *not* be allowed to use books. You will also *not* be allowed to use electronic devices, including calculators, cell phones, computers, ebook readers, Google glasses, smart watches, and tablets (this may not be a complete list). If you have questions about what is (or is not) allowed, then please ask me.

If you know that you will miss an examination, then you must contact me well before that examination to schedule an alternate date. If you miss an examination without contacting me in this way, then you must submit a written statement (such as a doctor's note) that adequately explains your absence before you can schedule an alternate date. Explanations like "I forgot there was a test" or "I overslept" or "my car wouldn't start" will not be accepted. Examinations given on alternate dates may be different from the others: they may be easier or harder.

Incompletes.

Incompletes will be granted only in rare instances, in which (1) you have completed all the coursework to date, and (2) an unforeseeable event prevents you from completing a small part of the remaining coursework. Incompletes will *not* be granted for foreseeable events, such as a heavy course load, or poor performance. You must request an incomplete as soon as possible, and you must provide a written statement (such as a doctor's note) that adequately explains why it is necessary.

Conduct.

Cheating on any part of this course is a serious offense. The consequences of cheating may include a score of zero points on any work in which the cheating occurred, and/or a failing grade in the course. It may also be reported to the Office for Student Conduct and Academic Integrity. More information is available at the CSci **Academic Conduct Site.**

Examples of cheating include (1) copying work from other students, (2) letting other students copy your work, (3) getting help from websites, (4) getting help from people other than me or the course TA's, and (5) using posted or common solutions (this may not be a complete list). If you have questions about what is (or is not) allowed, then please ask me.

Disabilities.

The University of Minnesota is committed to providing equitable access to learning opportunities for all students. The Disability Resource Center (DRC) collaborates with students with disabilities to provide for and arrange reasonable accommodations.

If you have, or think you may have, a disability (*e.g.,* mental health, attentional, learning, chronic health, sensory, or physical), then please contact DRC at 612 626–1333 to arrange a confidential discussion regarding equitable access and reasonable accommodations. More information is available at the **DRC Site.**

If you are registered with DRC and have a current letter requesting accommodations, then you may contact me early in the semester to review how the accommodations will be applied in the course.

Peer-Assisted Learning.

The **Peer-Assisted Learning** (PAL) program provides free weekly study sessions for CSci 2041 students. These are led by Roman Grebin (`grebi011@umn.edu`), an undergraduate who has taken the course and has been trained as a PAL facilitator. PAL sessions meet on Mondays from 4:00 PM–4:50 PM in **Amundson Hall** 124, and on Wednesdays from 10:10 AM–11:00 AM in **Keller Hall** 1–250.

Schedule.

Topics discussed on each date are approximate. I may change them during the course. Dates for programming projects do not appear on the schedule, but will be announced in class. There is no lab scheduled for the first week of class.

LECTURE DATES	LECTURE TOPICS	LAB DATES
Sep 7	Functional languages. History. Lisp. Clojure.	
Sep 14	More about Clojure.	Sep 13
Sep 21	Even more about Clojure.	Sep 20
Sep 28	Types.	Sep 27
Oct 5	More about types.	Oct 4
Oct 12, 6:30–7:30 PM	**First Midterm Examination.**	
Oct 12, 7:30–9:00 PM	Higher-order functions. Continuations.	Oct 11
Oct 19	Macros.	Oct 18
Oct 26	Interpreters. Metacircular evaluators.	Oct 25

Nov 2	Eager and lazy evaluation.	Nov 1
Nov 9	Infinite data objects.	Nov 8
Nov 16, 6:30–7:30 PM	**Second Midterm Examination.**	
Nov 16, 7:30–9:00 PM	Modularity.	Nov 15
Nov 23	Concurrency and parallelism.	Nov 22
Nov 24, 25	**University Closed.**	
Nov 30	Search-based programming.	Nov 29
Dec 7	Other languages. Java. OCaml.	Dec 6
Dec 14	More about OCaml.	Dec 13
Dec 21, 6:30–8:30 PM	**Final Examination.**	

The final examination will be held on a Wednesday. It will be in the usual **Keller Hall** 3–111 classroom.