

CSC151 Functional Problem Solving with Scheme

(Spring 2009)

Synopsis: This class introduces “computational thinking” as a method of solving problems. Using visual media (art and images) as a vehicle for exploration, you will explore important concepts from computer science that can be applied to a variety of fields. With daily labs that help you become progressively fluent in a high-level programming language, you will practice the design and analysis of algorithms, recognizing and creating abstractions, and using recursion. You also get to create some fun pictures, to boot.

Instructor: Jerod Weinman
Office: Noyce 3825
Phone: x9812
E-mail: [weinman]

Mentor: Max Kuipers
E-mail: [kuipersb]

Office hours:

Monday	3:00-4:00 PM
Tuesday	3:30-4:30 PM
Wednesday	4:00-5:00 PM
Friday	2:30-3:30 PM

or by appointment.

Course web page: <http://www.cs.grinnell.edu/~weinman/courses/CSC151/2009S/>
Class meetings: 9-9:50 am, MTWF, Science 3819

1 Overview

The official catalog description reads:

A lab-based introduction to basic ideas of computer science, including recursion, abstraction, state, information-hiding, and the design and analysis of algorithms. Includes introductory programming in a high-level, functional language. Prerequisites: None.

This means that you’ll be introduced to the basics of computation, learning how to adequately describe and decompose problems of a computational nature so that you can effectively tell a computer the steps it should take to solve the problem. We will study some beginning concepts that make this process possible, easier to undertake, and often elegant.

Our major objectives for this course include:

- Understanding some fundamentals of computer science: algorithms, data structures, and abstraction.
- Familiarity with the practice of computer programming (design, documentation, development, testing, and debugging) in a high-level programming language, Scheme.
- Learning problem-solving from a functional programming perspective.
- Sharpening general thinking and learning skills.

Why take it? Some basic knowledge about solving problems computationally can be useful in careers involving EVERY major offered at Grinnell. (I invite you to challenge me with one if you are skeptical.) This course will outfit you with some versatile concepts that are applicable to many other languages. Plus, you get to be creative in many different ways, from problem solutions to works of art. Creativity is highly encouraged!

Important Warnings

Warning! Experience shows that CSC 151 exercises different parts of your brain than other courses (even math and science courses). Expect some difficult times, but have confidence that you can work through them and that you'll come out of the course with much more knowledge.

Warning! Like learning a foreign language, learning in this course is cumulative: New ideas often build on ideas from earlier in the course. If you feel like you've missed something important, please come talk to me ASAP.

Warning! Computers have no common sense or compassion. They are complex, and sometimes they do things we don't expect. *When things go wrong, don't blame yourself.* Ask me or a tutor for help.

In addition, we are using software (MediaScript) that has been developed at Grinnell and may still have some bugs lurking within. It is entirely possible that it may crash for no good reason.

2 Textbooks / References

There is no required textbook for this course, as the material has been written by Grinnell faculty and will be posted on the web. However, there are some useful textbooks and references you may want to bookmark or keep handy.

Books

Dybvig, Kent R., *The Scheme Programming Language* (Third Edition), MIT Press, 2003.

<http://www.scheme.com/tspl3>

An effective guide to learning the Scheme language.

Felleisen, Matthias, Flinger, Robert Bruce, Flatt, Matthew and Krishnamurthi, Shriram, *How to Design Programs*, MIT Press 2001.

<http://www.htdp.org>

A wonderful guide, particularly for later topics in the course like higher-order programming, re-factoring, and recursion.

References

Kelsey, Richard, Clinger, William, and Rees, Jonathan, editors, *Revised⁵ Report on the Algorithmic Language Scheme*. 1998.

http://www.cs.grinnell.edu/courses/Scheme/r5rs-html/r5rs_toc.html

<http://www.cs.grinnell.edu/~weinman/courses/CSC151/doc/r5rs.pdf>

The complete, concise definition of the Scheme programming language.

3 Schedule of Topics

The following is an approximate schedule of topics to be discussed during the course. See the web page schedule for details.

<i>Week</i>	<i>Topic</i>
1	Introduction
2	Graphics, procedures, and style
3	Numbers, images, and color
4	Conditionals, list iteration
5	Transforming colors/images
6	Recursion
7	Preconditions, local procedures

<i>Week</i>	<i>Topic</i>
8	Geometric art, files
9	Pixel maps
10	Vectors, analysis, data structures
11	Deep recursion, association lists
12	Higher-order procedures, search
13	Sorting
14	Summary and review

4 Assignments and Activities

This class is taught in a collaborative workshop style. Some days we will spend working through problems or concepts in an interactive fashion. Most days, you'll work on laboratory problems on the computer with other students. We will start many days with a quick introductory discussion and end with a reflective discussion. **Participation is KEY!**

To make class time most valuable for you, I do not plan to lecture on material that is covered in the reading. By reading beforehand, we can concentrate on areas of confusion and highlight the important concepts you will be experimenting with that day. Because our class time is limited, you should come prepared to each class.

Q: What does it mean to be prepared for class?

A: Preparation consists of

1. Check the schedule for the day's meeting to find out the topic.
2. Read the assigned material *before* class.
3. Come to class on time, with paper and something to write with, ready to participate.

Under a normal 16 credit load, I expect that you will spend at least (but hopefully more) 40 hours per week on your studies (class time, homework, and studying). Thus, you should plan to spend 10 hours/week on work for this course. With class time clocking in at $3\frac{1}{3}$ hours, you should have at least $6\frac{2}{3}$ hours/week left for the remaining activities.

4.1 Reading

Nearly, every day there will be a brief reading assigned. Before each class, you should check the class schedule for assignments. Reading the material may entail the following:

Overview You should look over the reading once to get an overview of the material to be covered.

In-Depth Next, read the material closely. Perhaps not everything will make sense at this point, but hopefully most things will. You should try some of the examples in the text as you read.

Final Notes After carefully reading the material, try making a few notes to yourself about what you think are the most important concepts being covered, as well as any questions you have.

These readings are short, but very important. Some details will make more sense as you try things out on your own in lab, but I am always happy to answer your questions.

4.2 Participation

Because much of our work in this course involves collaboration and discussion, you will be evaluated on your participation.

Participating in class involves:

- being present in class (physically and mentally)
- coming to class on time
- coming to class prepared
- asking questions when appropriate
- making positive contributions to class discussion, by volunteering and when called upon
- staying on task during lab exercises, and
- working effectively with your lab partner(s)

Students who regularly meet these criteria can expect to earn 90% (i.e., an A-) for their participation grade. I will reward students who regularly provide significant insights or guide discussion in productive ways with a higher participation score. Students who fail to participate regularly or who participate in counterproductive ways (e.g., by dominating the conversation or making inappropriate comments) can expect to earn a lower score.

One unexcused absence will have no effect on your participation score. (See the Attendance Policy, Section 6 below.)

4.3 Labs

Most class days will involve collaborative laboratory work. You may not always complete the laboratory assignments during class. It is very important that you finish these outside of class to be sure that you are engaging in all the material we cover. Like playing an instrument or speaking a foreign language, the only way to become proficient is to practice, practice, practice! You will be expected to carefully write up your solutions to some lab exercises.

Lab write-ups will be due at the beginning of the next class session. When a lab is to be written up, I will let you know before you start through the course web site.

You and your lab partner(s) may complete the write-up together or individually; you should decide which before you start. If you complete the lab on your own, acknowledge your partner in your writeup.

Your write-up should include your solution to each exercise in the lab. If the exercise directs you to write Scheme code, include your Scheme code. If the exercise directs you to find out what the output of some expression is, copy and paste the output from the MediaScript console. If the exercise asks a question or asks you to explain something, write an answer in English. You do not need to copy the entire lab into your write-up (in fact, please don't!), but include the problem number for each exercise.

Send your lab write-up to me in the body of an email (do not send an attachment) by the beginning of the class day. The subject of the email should be of the form "CSC 151-01 Lab: *Title of Lab*". In general, I will not reply to your email.

Lab write-ups will be graded on a binary scale, 0 or 1. You will earn a 1 if your write-up includes a solution or evidence of serious effort for each exercise in the lab, and a 0 if the write-up was not turned in or if some exercises are not attempted. If you were not able to complete some exercises because of their difficulty, you can still earn full credit by explaining where you got stuck and coming to talk with me as soon as possible.

In short, every diligent student should earn full credit for lab write-ups.

4.4 Homework

A homework assignment is due each Wednesday (when there is no project or exam). The homework is intended to let you learn and apply some new algorithms and ideas, rather than merely checking that you've learned the basic concepts (as the quizzes do). However, they are not intended to be huge undertakings. If you find the assignments are consistently taking longer than 3-4 hours, please talk to me.

I will give you instructions about what forms of collaboration are permitted on each homework assignment. Since I want to give you room to go above and beyond the requirements of the assignment, homework will be graded on a simple ternary scale:

- ✓+ Exhibits exceptional insight and/or craftsmanship
- ✓ Meets the requirements of the assignment.
- ✓- Does not meet the requirements of the assignment.

I expect most work to receive checks (✓). Of course, no credit will be given if nothing is submitted. The course schedule includes 9 homework assignments; your lowest homework grade will be dropped.

4.5 Quizzes

Each Friday, there will be a short, written quiz covering one or two key ideas from the previous week. Since lab work is done in pairs, the quizzes are intended to check each individual student's understanding of these ideas in a timely fashion (i.e., well before the examinations). If you (or the class as a whole) are missing a key concept, I want to revisit that concept as soon as possible so we can build on it in later lessons.

The course schedule includes 12 quizzes; your lowest quiz grade will be dropped. Since the goal of the quizzes is to check that you have learned basic skills, an answer that is basically correct will receive full credit, even if there are minor issues. For code, my rule of thumb is that the code is basically correct if I believe you could make it work in a few minutes at the computer (and it uses the required ideas). A partially correct answer will receive partial credit at my discretion.

If you require accommodations for these quizzes, please come talk with me as soon as possible. The first quiz will be the second Friday of the semester.

4.6 Project

The course will also include one more substantial project, in which you will have an opportunity to exercise your creativity in working on a program over a week. I will inform you of the grading rubric when the project is assigned on Wednesday, April 15. **Project due date: Wednesday, April 22.**

4.7 Exams

As opportunities for you to demonstrate your programming and design prowess, there will be three take-home exams, and an *optional* final exam.

Exam 1	Wednesday 11 February
Exam 2	Wednesday 8 April
Exam 3	Monday 4 May
Final Exam	Friday 15 May (9 am)

You should find they challenge you to go beyond what we have done in class.

Take-home exams are *open notes*, *open computer*, and *open instructor*. However, because I intend the exams to assess your own individual understanding of the material, **collaboration on exams is not permitted**. You should not talk to anyone (except me) about take-home exams before they are due. You should not discuss the problems on the exam, nor your answers. You should not give any information

about your progress on the exam (e.g., which problems you have completed or which you found difficult), and neither should you ask others about their progress. If you have any doubt about what is and is not permissible, ask the instructor.

The final exam can be used to replace your lowest score on one other examination. Although the final for this class is optional, you may in fact wish to take it. Therefore I encourage you not to make airline reservations that will conflict with your final exam schedule.

The final will be an in-class exam. Talking with other students during the exam will not be permitted. You may ask me questions. The exam will be closed-book and closed-computer, but you will be allowed to use one double-sided, 8.5"x11" sheet of hand-written notes.

5 Grading

My goal is for everyone taking this course to be able to demonstrate familiarity, fluency, and excellence with the course concepts. I would be very happy if you all met the goals above and received "A"s. The following weighting of individual activities will provide a basis for evaluation,

Participation	10%
Homework	20%
Project	10%
Quizzes	20%
Exams	30%
Labs	10%

5.1 Extra Credit

Successfully undertaking an activity for extra credit will add 1/2 of a percentage point to your final score. I will propose opportunities for extra credit throughout the semester, and you may propose ideas as well.

5.2 Good-Faith Grade Guarantee™

Because I realize that computer science does not "click" with all students, I reward effort as well as outcome. Hence, students who make a good faith effort in this class will pass the class with at least a C+. A good-faith effort includes missing no more than two classes, turning in every homework assignment, and spending the requisite time on each examination.

6 Attendance Policy

I know that sometimes "things happen." Therefore, you will be granted **one** unexcused absence from class with no penalty for participation. However, this is an intensively collaborative lab-based course, and your presence is integral to your learning. Therefore, the following will be deducted from your participation grade:

2-4 absences	20%
5-7 absences	40%
8 or more absences	60%

If you wish me to acknowledge your absence as excused, you must either:

1. Notify me at least 7 days in advance to make suitable arrangements for your absence, *or*

2. Provide me with documentation of your absence's circumstances *post hoc* from Health Services or Student Affairs.

If you do miss a class, you must first talk to a classmate about any material that you may have missed. After that, you may follow up with me about any further questions or concerns.

You should complete the lab assigned for any days you are absent and be sure you understand the material. See the late policy below on missing class during homework or quiz days.

7 Academic Honesty

You, as students, are members of the academic community. Both the College and I expect the highest standards of academic honesty. (See the Grinnell College Student Handbook, e.g.,

<http://www.grinnell.edu/offices/studentaffairs/shb/section3/academichonesty>).

Among other things, this means clearly distinguishing between work that is your own, and work that should be attributed to others. It is expected that the collaboration policies given in this syllabus and on particular assignments will be followed. When you explicitly work as part of a group or team, you need not identify the work of each individual (unless I specify otherwise). Furthermore, any program results or output must be faithfully recorded, not forged. (A thoughtful explanation of unexpected behavior can often be a worthwhile submission and is much better than the alternative.)

You should never “give away” answers to homework assignments or examinations. You may, however, work together in developing answers to most homework assignments. Except as specified on individual assignments, each student should develop his or her own final version of the assignment. On written assignments, each student should write up an individual version of the assignment and cite the discussion. On non-group programming assignments, each student should do his or her own programming, although students may help each other with conceptual design and later debugging. When working on examinations, you should not use other students as resources.

As an instructor, I will meet my obligation to bring any work suspected to be in violation of the College's Academic Honesty Policy to the attention of the Committee on Academic Standing, after which there is no recourse with me.

8 Deadlines

Work is due at the time and date specified in the assignment. Each weekday (MTWThF) your work is late will reduce your grade by one level (exams excluded). Work must still be submitted by the due date if you have arranged a prior excused absence.

Because I am concerned about your health and well being, I will also accept late work if

1. you start the assignment at least three days in advance of the due date;
2. you get to sleep by midnight the night before the assignment is due;
3. you expend a reasonable amount of effort to complete the assignment by midnight;
4. you turn in a letter attesting to facts 1, 2, and 3 when the assignment is due; and
5. you talk to me ASAP about any problems you've had on the assignment.

Deadlines for assignments involving programming will automatically be extended by at least one class period if MathLAN is down for an unscheduled period of 3 or more hours during the two days preceding the assignment due date.

Absolute deadline: All homework must be submitted by Friday, May 8 at 5 p.m.

9 Getting Help

The Math Lab makes tutors for 151 available for help in the open laboratory, SCI 3815. These tutors can be found at regularly scheduled times, which are posted on the front door.

You can also get help from me. Please come by during my office hours to discuss the course content, get any extra assistance, or just talk about how the course is going. Note that if multiple students have similar questions or issues, we may work together as a group. If you cannot attend a scheduled office hour, you may also email me to schedule an appointment; please include 3-4 possible meeting times so that I can pick one that works for me.

I enjoy getting to know my students, but I prefer to reserve office hours for academic matters. If you would like to have a more informal conversation, I would be delighted to accept an invitation to lunch (<http://www.grinnell.edu/offices/studentaffairs/chaplain/religiousandspiritualprograms>).

Email is also a reliable way to contact me, but please allow 24 hours for a response (except on weekends, when I do not regularly read email). You may also call me in my office (x9812) for more urgent matters (e.g., you will be missing a lab due to illness).

10 Accommodations

If you have any disability that requires accommodations, please meet with me right away so that we can work together to find accommodations that meet your learning needs. You will also need to provide documentation of your disability to the Dean for Student Academic Support and Advising, Joyce Stern, located on the 3rd floor of the Rosenfield Center (x3702).

Thanks to Janet Davis, Sam Rebelsky, and Henry Walker for many elements of the syllabus.