In Computer Science 161, “Imperative Problem Solving and Data Structures,” we continue our study of problem solving with the assistance of computer programs, examining problems that are best addressed through the imperative model of computation. In this model, the solution to a problem is expressed as a sequence of instructions to a processor that has access to a store (sometimes called a memory). The instructions that the processor can perform enable it to read values (from the keyboard or from files, for example), to save them in its store, to perform simple arithmetic computations with them, and to write out the results of those computations. Some processors also have access to sensors from which they can acquire information about their physical environment and to effectors by which they can directly influence objects in that environment. Our approach will be to find the instructions that can be combined to form a step-by-step method for solving a problem and to arrange for them to be carried out by the processor in exactly the right order.

To express these instructions, we’ll use the C programming language, which is well adapted to this model of computation. C is most often used in practice for large programs and procedure libraries in which execution speed is a high priority.

The class meets in Noyce 3815 at 9 a.m. on Mondays, Tuesdays, Wednesdays, and Fridays, beginning on Monday, January 25.

Readings

There is one required textbook: C Programming: A Modern Course, by K. N. King (New York: W. W. Norton and Company; second edition, 2008; ISBN 978–0–393–97950–3). Some of the readings will be taken from this book; others will be provided as handouts.

I have asked the College Book Store also to make available C: A Reference Manual, by Samuel P. Harbison and Guy R. Steele, Jr. (Englewood Cliffs, New Jersey: Prentice-Hall; fifth edition, 2002; ISBN 978–0–13–089592–9). It explains the various constructions and built-in functions that every implementation of C must provide. The semantic rules of C are more intricate than those of Scheme and provide many more opportunities for programmers to make confusing errors, so you are likely to need to consult a standard reference frequently.

The instructor and the mentor

My name is John David Stone. My office is Noyce 3829. My office hours this semester are Tuesdays from 2 to 4 p.m., Thursdays from 9 to 11 a.m., and Fridays from 2 to 4 p.m., and by appointment. I can be reached by telephone at extension 3181.

If you use PGP encryption, you can also send me e-mail as reseda@grinnell.edu. My PGP public key is available on the Web at http://www.cs.grinnell.edu/~stone/pgp-public-key.txt

If you haven’t yet started encrypting your e-mail, the Free Software Foundation’s guide, Email Self-Defense (at https://emailselfdefense.fsf.org/) explains how to do so.

Our student mentor for the course, who will also assist me during class sessions, is Toby Baratta. She can be reached by telephone at 561–430–8098 (until 10:30 p.m.) or by e-mail at barattat17@grinnell.edu.
Requirements

Each student in the course is expected to study the assigned readings carefully, to learn the ideas, methods, and techniques presented there, to submit solutions to exercises requiring the application of those ideas, methods, and techniques, to prepare for and attend the sessions of the class, and to submit to two examinations, a one-hour in-class examination at mid-semester and a three-hour comprehensive final examination at the end of the course.

In the schedule of topics for the course, I have specified a reading assignment between successive class meetings. Please read each assignment before the beginning of the class that follows it in the schedule.

Labs and projects

This is a lab-based course, and the main activity at most of our class sessions will be to work on short, illustrative, tightly focused lab exercises. You will work on these labs in groups of two (or occasionally three) persons, a practice called “pair programming.” It is usual for one person to be logged in and managing the keyboard and mouse, writing up the solutions to that day’s exercises, while the other person acts variously as a source of ideas and suggestions, a skeptical observer and critic, a prompter, a motivator, or a research assistant, as appropriate. The idea that working together on an exercise helps both of the participants to understand their solutions better than if they worked alone, and your goal as students is to reach that better shared understanding, not just to follow the shortest path to the completion of the exercise.

At some of our sessions, you will be working instead on the eight projects that make up the backbone of the course. These projects involve writing programs that will be executed by the small car-like robots, so that they will interact with their environments by reporting data from their sensors, carrying out computations on those data and sending signals to their “effectors”—their motors and sound generators—to cause the robots to behave in ways that your programs specify.

These project programs will require more thought than you’ll be able to give them in one fifty-minute class session, so you and your project partner will also need to meet outside of class to work up your solutions and test them properly. These are the principal assignments for the course, and I will expect them to be your best work.

Class attendance

It would be a good idea to attend every session of the class and to participate actively. It is especially helpful if you raise for discussion any questions you may have about the day’s topic, the assigned reading, or the exercises. I suggest that you write out such questions as part of your preparation for class sessions and pose them as opportunities arise.

Notwithstanding that general expectation, class attendance is optional, (except for the examinations, which are obligatory). However, if you miss a class session for any reason, you must write up and submit solutions to the lab exercises for that session.

Days of religious observance

Grinnell College acknowledges and embraces the religious diversity of its students, faculty, and staff. If you miss a class session in order to fulfill a religious obligation, your absence is excused. The accommodation is the one described in the preceding section: You must write up and submit solutions for the lab exercises that you missed.
If one of your days of religious observance coincides with the final examination, please let me know early on in the semester so that we can arrange for you to take it at an earlier time.

In religious matters, I am a free thinker. I have no religious obligations that are tied to specific days, and I expect to attend all of the class sessions scheduled for this semester.

**The examinations**

The mid-semester examination for this course will be held at 9 a.m. on Monday, March 14 and the final examination at 2 p.m. on Thursday, May 19. Each examination will be an anthology of short-answer questions, problems (some involving calculation), and essay questions.

For the examination, you may bring and use any printed materials that you want to have available, but you may not use any electronic devices that are capable of computation, data processing, or communication.

**Grading**

Section IV.A.4 of the Faculty Handbook explains the duty of faculty members to report grades as follows:

Among the accepted responsibilities and obligations of each member of the faculty is that of reporting to the Registrar, at the appointed times, grades in accordance with the grading system and with the grading regulations which the faculty shall from time to time adopt. Further, faculty members are expected to make timely evaluations of students throughout the year.

Inherent in the responsibility of reporting grades is the further understanding that all such grades reported shall have been determined, in the final analysis, on the basis of the faculty members own professional evaluation of each individual students work.

This above indicated responsibility is considered to be part of the contractual relationship between the individual faculty member and the College, and the failure to fulfill this obligation will be considered a breach of contract.

In each of my courses, I fulfill this responsibility to the letter. However, I advise students that they should not regard the grades reported to the registrar as complete or even adequate assessments of their intellectual strengths and weaknesses. Grinnell College’s grading system is extremely inexpressive and vague.

The role of grades in the operation of the College is to support the generation of factoids (such as grade-point averages) that notionally provide quantitative justifications for decisions about progress towards graduation, off-campus study, academic honors, and such like. When collected on transcripts, grades also provide a convenient (though potentially misleading) way of summarizing academic performance for the benefit of personnel managers and graduate-school admission committees, who often cannot spare the time to read accurate assessments of candidates’ strengths and weaknesses.

I therefore urge you to pay much closer attention to your verbal interactions with me and to the comments that my assistants and I make on your course work than to the grade that I report at the end of the course. Nonetheless, that grade is, as specified, based entirely on my professional evaluation of your individual work.

Your performance on projects will determine two thirds of your final grade, class attendance and participation one sixth, your performance on the mid-term examination one twenty-fourth and your performance on the final examination the remaining eighth, with the condition that you must pass the final examination in order to pass the course.
Copyrights and licensing

Source code that I create for this course and make available on MathLAN in the /home/stone/c/code directory is licensed under the GNU General Public License. Code released under the GPL can be copied, studied, revised, improved, and redistributed freely, subject to the restriction that any copies or derived programs are released under the same license.

Similarly, handouts and other prose course materials that I have created and distributed either in hard copy or in /home/stone/c/handouts are licensed under the Creative Commons Attribution–ShareAlike 4.0 International License and can be copied, studied, revised, improved, and redistributed freely, provided that my authorship of the original work is acknowledged and that any copies or derived works are released under the same license.

Under Grinnell College’s copyright policy, a student who submits a program or a prose text to satisfy a requirement of this course retains the copyright to it. Similarly, a group that submits such a work has and retains a collective copyright. In either case, however, the College asserts the right to distribute the work within the College community for instructional or administrative purposes without paying any royalty to the student.

You may, of course, choose to publish your work under some more generous license, and I encourage you to use the GNU General Public License (version 3) for software and the Creative Commons Attribution–ShareAlike 4.0 International License for text.

To my knowledge, the College has never abused its claimed right of distribution or used it in a way that would be contrary to the interests of student authors. In 2013, however, the College designated Microsoft Corporation as a school official with a “legitimate educational interest” in student records. Microsoft Corporation has a much sketchier track record that includes many instances of exploitation of its customers and users. Its rap sheet includes a felony conviction, in American criminal courts, on charges of abuse of monopoly and a similar conviction in the European Union. In addition, since 2007, Microsoft has given the National Security Agency access to most of its users’ emails, video chats, and cloud document storage, in unencrypted form.

In my opinion, Microsoft cannot be trusted to respect the rights and interests of student authors. Accordingly, I now advise students not to store any original writing, including source code for computer programs, on the College’s cloud-based storage servers and not to e-mail it (without encryption) to or from any @grinnell.edu address, since either of those acts might be construed as allowing Microsoft, as a school official, to read, copy, modify, distribute, and mine data from your original work without your permission.

Collaboration and plagiarism

Since you will receive credit for this course on the basis of your individual performance, it would be unethical to submit the work of others as your own.

However, the labs and the projects for this course are collaborative in nature and are indeed designed to promote a collaborative approach to programming and problem-solving. When turning in projects, therefore, you should make sure that everyone who worked on the project signs off on it and explicitly takes both credit and responsibility for it.

Your answers to examination questions should be entirely your own. Collaborating on examinations is prohibited.

If I encounter clear indications of plagiarism or academic dishonesty, the Committee on Academic Standing will deal with them. The College’s policies for faculty members prohibit me from trying to investigate major offenses on my own. For the same reason, I
impose penalties for academic dishonesty only as directed by the Committee on Academic Standing.

Because in recent years the Department of Computer Science has received mixed signals from the Committee on Academic Standing and from the Office of the Dean on the nature and scope of academic dishonesty, we are currently in the process of formulating departmental standards of academic integrity for students in our courses. Although we have not yet published even draft versions for review and comment, it is possible that we will adopt some such standards during this semester.