

# CSC 161 Imperative Problem Solving and Data Structures

Fall 2017

**Synopsis:** This class broadens your means of “computational thinking” as a method of solving problems. Using robotics (sensing and control) as a vehicle for exploration, you will learn additional important concepts from computer science. With daily labs that help you become progressively fluent in a low-level programming language and its data representations, you will practice the design and analysis of algorithms, recognizing and creating abstractions, and manipulating important information structures. You also get to make robots sing and dance!

---

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

**Mentors:**  
**E-mail:**

**Course web page:** <http://www.cs.grinnell.edu/~weinman/courses/CSC161/2017F>

**Class meetings:** MTWF, Science 3815

- Section 1: 9:00-9:50 a.m.
- Section 2: 10:00-10:50 a.m.

## Contents

<b>1 Accommodations</b>	<b>2</b>
<b>2 Overview</b>	<b>2</b>
<b>3 Textbooks</b>	<b>3</b>
<b>4 Schedule of topics</b>	<b>3</b>
<b>5 Assignments and activities</b>	<b>4</b>
5.1 Reading . . . . .	4
5.2 Participation . . . . .	5
5.3 Module Laboratories and Projects . . . . .	5
5.3.1 Laboratory sessions . . . . .	6
5.3.2 Projects . . . . .	6
5.4 Short Quizzes . . . . .	7

5.5	Homework assignments . . . . .	7
5.6	Exams . . . . .	7
<b>6</b>	<b>Partner responsibilities</b>	<b>8</b>
<b>7</b>	<b>Grading</b>	<b>8</b>
7.1	Extra credit . . . . .	9
<b>8</b>	<b>Attendance</b>	<b>9</b>
<b>9</b>	<b>Academic honesty</b>	<b>10</b>
<b>10</b>	<b>Deadlines</b>	<b>11</b>
<b>11</b>	<b>Getting help</b>	<b>11</b>
11.1	Peer educators . . . . .	11
11.2	Discussion with Piazza . . . . .	11
11.3	Office hours . . . . .	11
11.4	Email . . . . .	11

## 1 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 Coordinator of Disability Services, John Hirschman, located on the 3rd floor of the Rosenfield Center (x3124).

Please also note that I require your accommodations. The chemical fragrances found in lotions, after shave, body sprays, scented laundry products, perfume, cologne, etc. make many people who suffer with asthma, allergies, environmental sensitivities, cancer, and migraines much sicker. I am sensitive to many chemicals you may not even notice, so please try to avoid using such scented products before coming to class and *especially* if you visit my office.

## 2 Overview

The official catalog description reads:

A continuation of CSC 151, bringing in some concepts more closely tied to the architecture of computers, compilers, and operating systems, such as macro processing, compilation and linking, pointers and memory management, data representation, and software development tools.

This section of CSC 161 will utilize robotics as an application domain in studying imperative problem solving, data representation, and memory management. Additional topics will include assertions and invariants, data abstraction, linked data structures, an introduction to the GNU/Linux operating system, and programming the low-level, imperative language C. The course will utilize a workshop style, in which students will frequently work collaboratively on a series of problems. Includes formal laboratory work.

This means that you'll be introduced to the C programming language, 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 more *fundamentals* of computer *science*: algorithms, data structures, and abstraction.
- Developing proficiency with the *practice* of computer *programming* (design, documentation, development, testing, and debugging) in a low-level programming language, C.
- Learning about *solving problems* in the imperative paradigm.
- Enhancing *general thinking* and learning skills.

Practically speaking, our topics will include:

- *imperative problem solving*: top-down design, common algorithms, assertions, invariants
- *C programming*: syntax and semantics, control structures, functions, parameters, macro processing, compiling, linking, program organization
- *data concepts*: data abstraction, integer and floating-point representation, string representation, arrays, unions, structures, linked list data structures, stacks, and queues
- *machine-level issues*: data representation, pointers, memory management
- *GNU/Linux operating system*: commands, bash scripts, software development tools

**Why take it?** Understanding another computational paradigm will increase your problem-solving abilities. Learning how your programs connect to the underlying representation and physical machine will give insight to their behaviors and limits. Oh, and programming robots is fun.

### 3 Textbooks

Students in this course should have a ready reference for the C programming language. Some labs include only a brief discussion of a topic, and students will need to do additional reading to understand the general context and the details of the material. Use of the textbook can be an effective means to study this material in appropriate depth.

K. N. King, *C Programming: A Modern Approach*, Second Edition, W. W. Norton, 2008, ISBN 978-0393979503.

The course web page provides a significant list of other important references.

### 4 Schedule of topics

We will work through eight primary modules, each with an integrative project. See the web page schedule for additional details.

**Module 0:** Getting Started: Linux terminal environment, C language, Scribbler 2 Robots, and Program Development (*Weeks 0–2*)

**Module 1:** Conditionals, Loops, and Robot Motion (*Weeks 2–3*)

**Module 10:** Arrays, Pointers, Functions, and Testing (*Weeks 4–6*)

**Data Representation:** Bits, Positive and Negative Integers, and Floating Point Numbers (*Week 6*)

**Module 11:** Characters, Strings, and Input/Output (*Weeks 7–8*)

**Module 100:** Structures and 2D Arrays (*Weeks 8–9*)

**Module 101:** Dynamic Memory Management, Linked Lists, and Program Management (*Weeks 9–11*)

**Module 110:** Abstract Data Types, Stacks, and Queues (*Weeks 11–13*)

**Module 111:** Bash Scripts, Command-line Arguments and File I/O (*Weeks –14*)

## 5 Assignments and activities

This class is taught in a collaborative workshop style. Some days we will work through problems or concepts in an interactive fashion. Most days you'll work on laboratory problems on the computer with another student.

To make class time most valuable for you, I do not plan to lecture on material that is covered in the reading. Instead, because experimenting and practicing is the best way to learn, you'll have the opportunity to ask questions and then begin working collaboratively on the day's lab exercises with the instructor and a class mentor available to provide assistance.

By studying the day's topics beforehand, we can concentrate the beginning of class on areas of confusion. Because our class time is limited, you must come prepared to each class, meaning you should:

1. Check the schedule for the day's meeting to find out the topic.
2. Study the assigned material *before* class.
3. Come to class on time, with your textbook, 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).<sup>1</sup> Thus, you should plan to spend a *minimum* of 10 hours/week on work for this course:

### 5.1 Reading

Nearly every day there will be brief readings assigned, some from the textbook and some from course web pages. No later than the night before each class, you should check the course schedule for the readings. Reading the material may entail the following:

**Overview** You should look over each reading once to get an overview of the material to be covered (e.g., summary and section headers).

**In-depth** Next, study the material closely. Perhaps not everything will make sense at this point, but hopefully many or most things will.

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

---

<sup>1</sup>This is a minimum recommendation for achieving “satisfactory” (i.e., C-level) results. “Good” or “excellent” results may require a greater investment.

These important readings are often brief, but dense, introducing non-trivial ideas. You should plan to spend an hour working on the material in the readings before class. At the end of the hour, you should be ready to discuss in class the material from the reading that is most important and most confusing to you. Thus, you should consider answering the following questions to be part of your daily preparation:

- Identify the section or concept from today's reading that you find most confusing.
- Briefly explain what you find confusing about it.

It is also extremely helpful to read the material no later than the night before class. Studies have shown that a little bit of forgetting between when you try to recall the information (like we will do at the beginning in class), can actually help you remember it *better* in the long run.<sup>2</sup>

## 5.2 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 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 (e.g., demonstrating a lack of preparation or involvement during lab exercises) or who participate in counterproductive ways (e.g., by dominating the conversation, making inappropriate comments, or getting off-task) can expect to earn a lower score.

**Portable electronic devices** Please refrain from using your phone, tablet, or other portable electronic devices during class. The distraction of e-mail, social media, messaging, and other similar platforms severely impede your learning and such activities are disrespectful to your partner. Repeated offenses will negatively impact participation grades.

An exception may be made at the conclusion of class, when you may wish to trade contact information with your partner or establish a schedule to meet for out-of-class collaboration.

## 5.3 Module Laboratories and Projects

Topics in this course are organized into eight modules and a few supplementary labs. Modules include laboratory sessions and conclude with an integrative project.

---

<sup>2</sup>Soderstrom, N.C., & Bjork, R.A. Learning versus performance, in Dunn, D.S. (ed.). Oxford Bibliographies in Psychology (New York: Oxford University Press).

### 5.3.1 Laboratory sessions

Most class days will involve collaborative laboratory work; I randomly assign in-class lab partners that rotate every module. Labs introduce specific features of Grinnell’s computing environment, highlight concepts and constructs introduced in class, allow instructor assistance in a “hands-on” setting, and supplement normal office hours.

You might not complete the labs during class, but it is important to finish these outside of class to be sure you are engaging in all the course material. Like playing an instrument or speaking a foreign language, the only way to become proficient is to practice, practice, practice!

Because of this, you may be expected to submit one or two exercises from each day’s lab, as announced at the end of class. For this reason, but most importantly for your learning, you should keep a careful record of your work as you progress through the lab.

As a special incentive for mastering the laboratory exercises, 30–50% of the problems on each test and on the final exam will be taken from the laboratory exercises (with only slight editing).

**Deadlines** When lab exercises are to be written up, I will let you know at the end of the class session in which it is conducted. You can view which labs have exercises due on the course web labs page. Lab write-ups are due by 9 am according to the following schedule:

<u>Lab Held</u>	<u>Writeup Due</u>
Monday	Wednesday
Tuesday	Friday
Wednesday	Monday
Friday	Tuesday

This schedule ensures you have time to ask questions of the instructor or tutors before the deadline. However, you are *strongly* encouraged to complete labs before the deadline (e.g., by the *next* class), because the next intermediate labs or homework may depend on them. **No late labs will be accepted.**

**Collaboration** You will work in class on your labs with a partner(s). After class, you and your lab partner(s) may submit your write-ups jointly (if you complete it together) or individually (if you complete it separately, outside of class). See the course web page for more details on format and submitting

**Grading** 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 on time or if some assigned 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 the instructor as soon as possible. In short, every diligent student should earn full credit for lab write-ups.

### 5.3.2 Projects

Each of the eight modules concludes with a project involving a computer program and associated commentary. These projects require you to integrate the material from the module by meeting several somewhat open-ended specifications.

**Deadlines** Deadlines for the projects vary throughout the semester and are posted on the course schedule.

**Collaboration** Collaboration on projects is **required**.

**Grading** All projects are evaluated under the general rubric for the course; additional specific grading criteria are applied to each project individually.

## 5.4 Short Quizzes

At the end of each module, we will have a very short quiz to help practice recall for the later exams and upcoming project. These will take just five minutes at the beginning of class and will be short answer, fill-in the blank, and/or multiple choice questions designed to help reinforce your learning of important elements from the module. Quizzes count for relatively little in your overall grade and are not designed to cause significant stress. Rather, they should help you *learn*. Even if you don't ace every quiz, studies show that just taking it will help you learn more in the long run, which can improve your exam scores.<sup>3</sup>

The following are the tentative quiz dates, scheduled either the same day as the project or the class day before.

Module	Week	Date
0	2	Tue 5 Sep
1	3	Wed 13 Sep
10	6	Mon 2 Oct
11	8	Tue 24 Oct
100	9	Wed 1 Nov
101	10	Fri 10 Nov
110	12	Wed 22 Nov
111	14	Tue 5 Dec

No make up or advance quizzes will be scheduled. The two lowest quiz scores will be dropped.

## 5.5 Homework assignments

Homework assignment problems extend the range of problems considered in the course and help sharpen problem-solving skills. To support this objective, **all homework problems are to be done individually.**

You may ask the instructor about any part of the course (including any homework problem) at any time. However, you must not discuss any homework problem with other students (e.g., students from the class, CS majors, or other students). Allowable help from peer educators is limited. See Allowable Help from Peer Educators

**Deadlines** Deadlines for the homework vary throughout the semester and are posted on the course schedule.

**Collaboration** Collaboration on homework is **forbidden**.

**Grading** All assignments are evaluated under the general rubric for the course; additional specific grading criteria are applied to each homework assignment individually.

## 5.6 Exams

As opportunities for you to demonstrate your grasp of the course material, there will be three midterm hour exams, and a final exam, all in-class.

---

<sup>3</sup>Roediger, H.L., & Karpicke, J.D. (2006). Test-enhanced learning: Taking memory tests improves long-term retention. *Psychological science* 17(3), 249–25.

Exam	Week	Date
1	4	25 September
2	7	13 October
3	11	13 November
Final	-	
Section 01		Tue 12 Dec, 9 am
Section 02		Fri 15 Dec, 9 am

## 6 Partner responsibilities

Work on labs and projects in this course is often done collaboratively (in pairs, occasionally in a group of three). Many studies suggest substantial benefits to learning with this type of group work. However, to be successful, collaboration requires partners to participate actively .

As noted above, partners should actively participate throughout class sessions and make arrangements to meet in the lab outside of class to finish labs or projects, as needed. Each partner has an obligation to attend and contribute during planned out-of-class sessions.

Failure to meet one’s responsibilities to a group negatively impacts the individual and impedes the partner’s education. Thus, except in exceptional circumstances (e.g., illness, family emergencies, serious injury), failure to follow through with one’s responsibilities as a partner may have a significant impact on one’s course grade and/or one’s standing in the course.

For more details and suggestions on how to be an effective pair programmer, please read the following article.

Wray, S. (2010). How pair programming really works. *IEEE software*, 27(1), 50–55.

## 7 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.

Discussion and participation	5%
Lab writeups	10%
Homework problems	15%
Projects	15%
Midterm Exams (3)	30%
Final exam	15%
Best of homework and midterms	5%
Quizzes	5%

Some work may be graded by someone other than the instructor. However, any questions or concerns about grading should only be directed to the instructor.

The following is proposed as the grading scale for the course.

Average %	Receives	Grade Points	Definition
93-100	A	4.00	Excellent
88-92	A-	3.67	Excellent
85-87	B+	3.33	Good
81-84	B	3.00	Good
78-80	B-	2.67	Good
74-77	C+	2.33	Satisfactory
68-73	C	2.00	Satisfactory
55-67	D	1.00	Passing
0-54	F	0.00	Failing

To compensate for the unpredictability of learning exercises' outcomes, the brackets (left column) *may* be adjusted upward (but not downward).

## 7.1 Extra credit

Computer science is a wide-ranging discipline, and courses can cover only selected pieces. To encourage students to expand their horizons, students may earn 2 (unnormalized) points extra credit for each *Thursday Extra* or other departmental talk attended, with 1 point extra credit counted toward homework problems and 1 point extra credit counted toward lab writeups. Additional extra credit opportunities may be announced through the semester.

**Content** To earn this extra credit, you must send me an email with a paragraph or two (plain text, please, no attachments) summarizing what you learned. Do not send a summary of the content; I find reading your reflections on the content more interesting than reading a simple recapitulation.

**Deadline** To receive credit, you must submit your reflection *within one (1) calendar week of the event*.

**Limits** Extra credit from any source may only raise your final grade at most one bracket. For example, if your grade average without extra credit is 79% (a B-), but your extra credit raised your average to 85% (a B+), your final grade would be capped at a B. Thus, you should plan to do as well as possible as the semester progresses.

## 8 Attendance

Because is a collaborative, discussion-based course, your presence is integral to your learning. Thus, 1.5% will be deducted from your *overall* grade for each unexcused absence. I know that sometimes “things happen.” Therefore, you will be granted **one** unexcused absence from class without penalty. However, this rebate is cancelled upon a second absence.

**Tardiness is indistinguishable from absence** (until you arrive), and this causes problems in ensuring everyone has a partner for the day's work. Therefore, if you are more than five minutes late to class or arrive *after* we start the lab (when I will have needed to resituate your partner) you will be marked absent. Please plan to arrive by the start of class.

If you will be absent, you must send a written explanation (email is acceptable) **before** class except in the (rare) case of dire emergency. (Consider the analogy for phoning in sick for work; you wouldn't do it at 4 PM.)

*Because I do not wish you to risk harm to yourself or others, I am likely to moderate penalties in case of illness.*

If you know in advance that you will be absent for any reason, you must notify me in writing (again, email is fine) at least 7 days in advance to make arrangements for considering your absence excused.

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.

## 9 Academic honesty

As students, you are members of the academic community. Both the College and I expect the highest standards of academic honesty, as explained in the Grinnell College Student Handbook,

[http://catalog.grinnell.edu/content.php?catoid=12&navoid=2537#Honesty\\_in\\_Academic\\_Work](http://catalog.grinnell.edu/content.php?catoid=12&navoid=2537#Honesty_in_Academic_Work).

Among other things, this means clearly distinguishing between work that is your own, and work that should be attributed to others. This includes ideas, examples, and code that you draw from labs and readings.

It is expected that the collaboration policies given in this syllabus, on particular assignments, and in the allowable help from peer educators will be followed. In particular:

- When you explicitly work as part of a group or team, you need not identify the work of each individual (unless I specify otherwise).
- You may discuss *concepts* (algorithms, ideas, approaches, etc.) described in the readings, lab exercises, or during class with *anyone*.
- You may only discuss course assignments (algorithms, solutions, write-ups, code, debugging, etc.) with your group members, computer science tutors, CSC 161 mentors, or the CSC 161 instructors.
- All the work submitted (code, experimental data, write-ups, etc.) must be your own or that of your group. **Code or documentation provided by the instructor must be attributed**, including code that you copy and subsequently modify.
- All non-syntax consultations (i.e., ideas about algorithms) from any source, including the readings, labs, provided code, and internal or external language references, require formal citation within the related program or write-up.
- **Any conceptual contributions by individuals not in your group must be acknowledged and attributed in your report.** That is you must give specific attribution for **any** assistance you receive. (This includes assistance from tutors or mentors.) The suggested acknowledgment format is

“*[Person X] helped me to do [thing Y] by [explaining Z].*”

- 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 are responsible for safeguarding your work from being copied by others. This requires you to take reasonable precautions with hard copy printouts as well as file system permissions. (Note that MathLAN’s default permissions prevent others from viewing your files.)

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 can be no recourse with me.

## 10 Deadlines

Work is due at the time and date specified in the assignment. Unless otherwise specified, the deadline for a project or homework problem is 10:30 PM on the date specified.

Assignments due on days for which you have a prior excused absence must still be submitted by the deadline.

A late penalty of one letter grade will be deducted in each subsequent twenty-four hour period after the deadline.

*Exception: 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 work (except the final) must be submitted by Friday 8 December at 5 p.m. (Note that this is earlier than the institutional deadline).

## 11 Getting help

### 11.1 Peer educators

Our course mentors will hold weekly evening mentor sessions to go review, connect, and/or go further in-depth on the course topics.

The Computer Science Department makes tutors for CSC 161 available for drop-in help in the open laboratory, SCI 3815 (Sunday-Thursday, 7–10 PM). Peer tutors may also be available for regular, more intensive one-on-one tutoring. As the course gets underway, please let me know if you are interested in regular weekly individual tutoring. I may also recommend individual tutoring.

### 11.2 Discussion with Piazza

For online class discussion or Q&A we will use Piazza, which is designed to get you help fast and efficiently from your classmates, mentors, and myself. Rather than emailing questions, please to post your questions on Piazza: <https://piazza.com/grinnell/fall2017/csc161/home>. You must ensure your queries and responses respect the academic honesty and collaboration policies for the course and specific problem.

### 11.3 Office hours

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.

Half of my office hours are open, “drop-in” times for students with quick or unplanned questions. If there is anyone waiting, I try to will limit our time to ten minutes to respect others’ time. The other half may be reserved by signing up for a 15 minute slot on the bulletin board outside my office.

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 eat lunch with you at the Marketplace.

### 11.4 Email

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).

---

*Thanks to Janet Davis, Sam Rebelsky, and Henry Walker for many elements of this syllabus. Henry Walker wrote the Textbook (3) section, the topics list of the Overview (2), and most of the Module Laboratories and Projects (5.3), and partner responsibilities (6) in CSC 161: Imperative Problem Solving and Data Structures under a Creative Commons Attribution-NonCommercial-Share Alike 4.0 International License.*

---

Copyright ©2017 Jerod Weinman.



This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 4.0 International License.