COLORADOSCHOOLOFMINES
EARTH • ENERGY • ENVIRONMENT

CSCI-400: Programming Languages (PL)

Term: Spring 2022
Instructor: Dr. Neil T. Dantam

Contents

1 Overview and Outcomes

Previous courses have examined how to write programs in individual languages such as Python or C++. In this class, we will take a broader view and study the key concepts and techniques that allow developers to design and implement programming languages. Ultimately, the course will improve your skill as a programmer and will deepen your understanding of how programming languages are designed and implemented.

Some students expect a "Programming Languages" course to be a survey of many different languages, but that is not the case. Learning "Programming Languages" is less about touring of the "zoo" of different languages and more about learning the "zen" that underlies all programming languages. Ultimately, learning these common and fundamental abstractions is the more valuable approach. We must all learn and use many different languages over our careers; understanding the fundamentals of programming languages will make this task easier and make us better programmers.

This course will emphasize *functional programming* for several reasons:

- Functional programming generalizes many programming constructs you have previously learned;
- Functional programming predicts the future of language development. Many programming languages that began in another style (imperative, object-oriented, etc.) are gradually gaining functional programming features over subsequent revisions;
- And functional programming helps us write correct (bug-free) programs. Functional languages offer many tools to reason about and ensure program correctness, both informally (in our heads) and formally (with an algorithm).

Through the activities in this course, you will learn the following (Figure 1):

Remember definitions of conventional object in programming language theory.
Example: Define a lexical closure

Understand functional programs.
Example: Use the `map-reduce` idiom.

Implement the core of a functional programming language.
Implement an expression evaluator.

Analyze Trade-offs among programming approaches.
Example: What are the pros and cons of a particular style or language for a given application?

Evaluate the suitability of various programming approaches for new problems.
Example: Explain why should we use a particular style for a given application.

Create formal proofs.
Example: Prove the amortized running time for a data structure or algorithm.

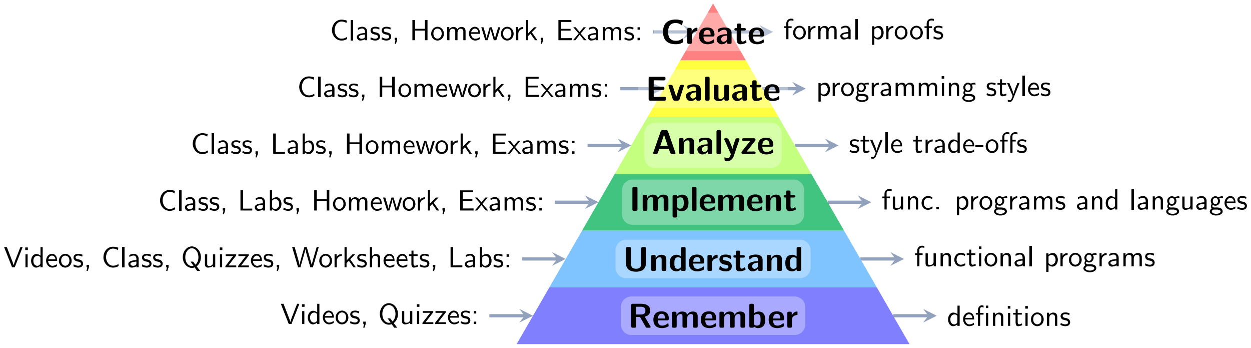


Figure 1: Bloom's Taxonomy of Learning Activities and Outcomes

2 General Course Information

Instructor: Dr. Neil T. Dantam
TAs/Graders: Kieran Lewis, Christopher (Chris) Muller, Grant Rulon, Aiden Sweezey, Zabdiyel (Zab) Tan

Prerequisites The prerequisites are CSCI-358: Discrete Math and CSCI-306: Software Engineering. If you have somehow registered without completing the prerequisites, please drop the course.

Textbooks and References

- **Primary Textbook** Clarkson et al. OCaml Programming: Correct + Efficient + Beautiful.
- **Supplemental References**
  - Chris Okasaki. *Purely Functional Data Structures*. Cambridge University Press. 1998. ISBN-13: 978-0521663502.
  - Chris Okasaki. Purely Functional Data Structures. Ph.D. Thesis. CMU-CS-96-177. Carnegie Mellon University. 1996. (A "draft" version of the above book)
  - Benjamin Pierce. *Types and Programming Languages*. 2002. ISBN-13: 978-0262162098
  - Matthew Hennessy. *The Semantics of Programming Languages*.
  - Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman. *Compilers: Principles, Techniques & Tools*. ISBN-13: 978-0321486813.
- **Programming References**
  - OCaml Tutorial
  - OCaml Manual
  - JavaScript Specification
  - Mozilla Developer Network

Office Hours

- Instructor office hours will be both face-to-face (BB280H) and on Zoom (see Piazza for the link).
  - TBD
- TA Office Hours are remote. (see Piazza/Google Calendar for the link).
  - TBD

Online Resources

- Canvas: Grades
- Piazza: Announcements, Questions, Discussion, Homework/Project help
- MSOneDrive: Files
- Course Github Organization: Project code distribution and submission

- Isengard: ITS-managed Linux server with shell access
- MS Whiteboard: In-class discussion
- Zoom: Remote office hours

Technology Requirements This course assumes you are able to access a GNU/Linux system (e.g., Debian, Ubuntu). If you do not run Linux on your personal workstation, you may use the ITS-managed Isengard server or install a virtual machine. The instructor does not recommend Microsoft's "WSL" due to problems encountered by past students. If you attempt to use a non-Linux platform, the instructor and TAs can provide only very limited technical support.

You will need to bring a device to class for coding, quizzes, and (electronic) discussion. On days involving programming, this device must be a laptop with Linux access. On other days, a mobile device with a web browser will be adequate to complete quizzes on Canvas and post discussion notes. Please contact the instructor if bringing a device to class would present difficulty for you.

Who should I email/contact?

- **Miscellaneous basic policy questions:** (when is the midterm? when is an assignment due?): Re-read the syllabus, check Canvas assignments, check Piazza announcements, and ask any additional questions on Piazza.
- **Help with assignments or course topics:** Piazza, TA office hours, or instructor office hours. Private post on Piazza if the matter should be hidden from other students (e.g., something about your code or questions about your grade)
- **Solutions to in-class exercises:** Slides with completed exercises will be posted to MSOneDrive after the lecture.
- **Anything sensitive or confidential:** (e.g., a health issue) Email the instructor about the issue and/or to schedule a meeting to discuss the issue.
- **Concerns/suggestions about course procedures:** Email the instructor or TAs about the issue and/or to schedule a meeting to discuss the issue.

3 Grading and Evaluation

The course score (percentage) will be computed as a weighted average of scores (points received over points possible) as follows:

Participation	10% (w)
Quizzes	5% (q)
Homeworks	10% (h)
Labs	45% (ℓ)
Midterm Exam	13% (m)
Final Exam	17% (f)

$$\overbrace{s}^{\text{Course Score}} = .10\left(\frac{\mathtt{w}_{\text{recv.}}}{\mathtt{w}_{\text{poss.}}}\right) + .05\left(\frac{\mathtt{q}_{\text{recv.}}}{\mathtt{q}_{\text{poss.}}}\right) + .10\left(\frac{\mathtt{h}_{\text{recv.}}}{\mathtt{h}_{\text{poss.}}}\right) + .45\left(\frac{\ell_{\text{recv.}}}{\ell_{\text{poss.}}}\right) + .13\left(\frac{\mathtt{m}_{\text{recv.}}}{\mathtt{m}_{\text{poss.}}}\right) + .17\left(\frac{\mathtt{f}_{\text{recv.}}}{\mathtt{f}_{\text{poss.}}}\right)$$

Lecture Participation Most lectures will have a worksheet to practice the material. Scan or photograph the worksheet (or complete electronically) and submit it on Canvas. Your participation grade will be based on making an honest effort on the exercises. Participation points may also be assigned for in-class activities.

Quizzes Most lectures will include a quiz, graded on correctness.

Exams There will be a midterm exam around the middle of the semester and a cumulative exam during finals week. The exams are an evaluation tool whose purpose is to produce a distribution of scores across all students that best distinguishes the extent of learning. Thus, please expect the exams to be very challenging.

Homeworks There will be several homeworks sets to practice mathematical topics covered in the course.

Labs There will be several programming projects (labs) to practice functional programming use and implementation.

Letter Grades Course letter grades will be based on a curve. It is expected—but not guaranteed—that score distributions will be normally distributed and letter grades will correspond to university and department norms. However, skewed student effort (including absenteeism) or score distributions may result in correspondingly skewed letter distributions. See Appendix B for score distributions in prior terms.

Additional Letter Grade Criteria In addition to the scoring and letter grade determination discussed above, the following criteria will be used to determine letter grades:

1. If less than 80% of participation assignments are attempted and submitted on time, the course grade will be reduced by one letter, to a minimum of D-. This requirement is to encourage timely participation and "keeping-up" with the course topics.
2. If less than 80% of of quizzes are attempted and submitted on time (irrespective of correctness), the course grade will be reduced by one letter (cumulative with participation), to a minimum of D-. This requirement is to encourage attendance and participation in the class meetings.

Late Policy Late work will not be accepted. Please take care to manage your time so that you are able to submit your best work by the deadline.

Fairness It is import to evaluate all students as evenly as possible. While we will attempt to accommodate disabilities and extenuating circumstances (physical/mental health, school-related travel, job requirements of self-supporting students, etc.) to the greatest possible extent, it would be unfair to offer any further special treatment.

Grading Corrections Grading changes will only be made for grading errors. It is not possible to change grades in response to disagreements about point allocation, partial credit, letter grade cutoffs, etc., because such changes would be unfair to the rest of the class. Grading corrections will only be made for the following errors:

1. *Arithmetic:* The grader incorrectly summed your points.
2. *Code:* An error in the grading environment or scripts incorrectly tested your code.
3. *Written:* The grader incorrectly understood your answer.

Projects Expectations and Grading

- Code must produce the correct output to receive credit. Incorrect output, no output, compilation errors, or runtime errors will not receive credit. **Please double-check your submitted code to ensure that minor errors will not result in major test failures.**
- Code tests will include edge cases. Think through all possible conditions for your program.

Written Work Format and submit your written work as follows. Improper submission or formatting may result in a penalty on assignments.

- For FERPA compliance, all work submitted on physical paper must include a cover sheet that contains only your name and no answers or other work. Electronic submissions do not need a cover sheet.
- Write your name on *every page* of all written work. If the work cannot be matched to you, you cannot receive credit for it.
- Handwritten work must be *clearly legible* to receive credit.
- Submit electronic homeworks, reports, etc. in PDF format. Do not submit word processor files because these are inconsistently formatted by different software.
- Work must be readable when printed in black and white.

4 Tentative Schedule

Sure to change as the semester progresses. Updated 2022-06-11.

W	Date	Topic	Files	References
Part I: Functional Programming				
1	Jan 11	00: Programming Languages Introduction	📝	Clarkson 1
	Jan 13	01: Basics of Programming Languages	📝 📼	Clarkson 2, Pierce 5.1
2	Jan 18	02: Programming with Lambda	📝 📼	Pierce 5.2
	Jan 20	03: Linux and Git		Git Docs
3	Jan 25	04: Functional Programming	📝 📼	Hennessy 2, Okasaki 1–2
	Jan 27	05: OCaml and Javascript		OCaml tutorial & ref., JS spec, MDN
4	Feb 1	Career Day (no class)		
	Feb 3	06: Higher-Order Functions	📝 📼	Clarkson 4
5	Feb 8	Lab Day (Lab 2)		
	Feb 10	07: Algebraic Data Types	📝 📼	Clarkson 3, Pierce 11
6	Feb 15	08: Persistent Data Structures	📝 📼	Clarkson 5.6, 8.3, Okasaki 2–3
	Feb 17	Lab Day (Lab 3 & 4)		
7	Feb 22	President's Day (no class)		
	Feb 24	Midterm Review (tentative)		
Part II.a: Implementing a Functional Language, Syntax and Parsing				
8	Mar 1	Midterm (tentative)		
	Mar 3	09: Syntax	📝 📼	Clarkson 9.2, Hennessy 1.2, Aho 4
9	Mar 8	10: Lexing	📝 📼	Hennessy 1.2, Aho 3
	Mar 10	11: Parsing	📝 📼	Aho 4.4, 4.5, 4.8
Part II.b: Implementing a Functional Language, Typing & Evaluation				
10	Mar 15	Lab Day (Lab 5)		
	Mar 17	12: Induction	📝 📼	Clarkson 6.7, 6.8, Pierce 2.4, 3.1–3.3
11	Mar 22	Spring Break (no class)		
	Mar 24	Spring Break (no class)		
12	Mar 29	13: Semantics	📝 📼	Clarkson 9.3, Pierce 3.4–3.5
	Mar 31	14: Environments and Scope	📝 📼	Clarkson 9.4, Pierce 3.4–3.5, 7
13	Apr 5	15: Typing	📝 📼	Clarkson 9.5, Pierce 8–9, 22
	Apr 7	Lab Day		
14	Apr 12	16: Type Inference	📝 📼	Clarkson 9.6, Pierce 8–9, 22
	Apr 14	Lab Day		
15	Apr 19	17: Mutability	📝 📼	Clarkson 7, Pierce 11.2–11.3, 13
	Apr 21	Lab Day		
Part III: Extra Bits				
16	Apr 26	18: Amortized Analysis	📝 📼	Clarkson 8.2, Okasaki 5, Tarjan
	Apr 28	Lab Day		
17	May 3	Review		
	May 5	(Dead Day) Extra Review, in Green Center Petroleum Hall		
18	May 10	Finals Week		
	May 12	Finals Week		

5 Policies

5.1 Flipped Classroom

We will run this course as a “flipped classroom” to provide students with the additional exposure and activities supporting course learning outcomes. In particular, repeated exposure, study, and practice supports learning the formal and mathematical topics in this course.

- **Before Class,** please do the following:
  1. Watch the lecture video.
  2. Attempt the practice exercises (worksheets).
  3. Post lecture and exercises questions on the MS whiteboard to discuss during class.
  4. Submit the worksheet attempt on Canvas. Worksheets are graded on effort, not correctness.
- **During Class,** we will do the following:
  1. Answer questions about the video and worksheet exercises
  2. Discuss the lecture topics.
  3. Take a quiz
  4. We conduct additional activities such as coding exercises and group proofs. **Many in-class activities will closely resemble exam questions.**

5.2 Mines Policies and Resources

Mines Policies and Resources

5.3 CS Collaboration Policies

CS Collaboration Policies

5.4 Course Policies

5.4.1 Communication

Piazza is the primary communication tool used in this class and will be used for announcements, questions, and discussion. Students are expected to regularly monitor Piazza and their university email (at least once a day) for announcements and changes such as modifications to class meetings. The instructor will attempt to give at least 24 hours notice (and more if possible) for such changes, but emergency situations may not allow such advance notice. If in doubt, check your email and Piazza.

5.4.2 Laptop and Smartphone Policy

- Lecture slides are posted in advance. You are strongly encouraged to use your laptop or phone to follow along during lecture and to review slides during exercises.
- Note-taking on laptops, tablets, etc. is welcome if you find it useful.
- Please refrain from using laptops, phones, etc. for non-class activities, e.g., email, web browsing, games, during classtime, as it is distracting to other students.

- Some class activities (e.g., coding activities, lab days) require the use of a laptop. If you will not have access to a laptop for such activities, please contact the instructor about possible arrangements or alternatives.

5.4.3 Netiquette

Text DOs

- Ask questions and engage in conversations as often as possible—feel free to contact the instructor and TAs via the discussion forum for questions.
- When asking "tech support" questions, provide sufficient detail to diagnose and, if possible, reproduce the issue, including commands that were run, output of those commands, log files, and operating system and software versions.
- Be patient and respectful of others and their ideas and opinions they post online.
- Remember to be thoughtful and use professional language. Keep in mind that things often come across differently in written text, so review your writing before posting.
- Be prepared for some delays in response time, as "virtual" communication tends to be slower than "face-to-face" communication. Ask questions well in advance to deadlines to ensure sufficient time for a response.
- If the instructor does not respond to an important email for a few days, please send a reminder. Faculty receive a large number of emails, and sometimes messages get lost or overlooked.
- Contact the instructor if you feel that inappropriate content or behavior has occurred as part of the course.

Text DON'Ts

- Use inappropriate language—this includes, but is not limited to, the use of curse words, swearing, or language that is derogatory.
- Post inappropriate materials—for example, accidentally posting/showing a picture that is not appropriate for the course content.
- Post screenshots (images) of text output. Instead, post text as text. Compared to text, screenshots are slower to download, harder to read, and cannot be copy/pasted.
- Post in ALL CAPS, as this is perceived as shouting, and avoid abbreviations and informal language (e.g., "I'll C U L8R").
- Vent, rant, or send heated messages, even if you feel frustrated or provoked. Please instead communicate any specific concerns privately to the instructor or TAs; we want to improve the course and to accommodate any extenuating circumstances. Similarly, if you should happen to receive a heated message, do not respond to it.
- Except for course content questions on Piazza, send an email or post to the entire class, unless you feel that everyone must read it.

Video DOs

- Find a quiet place to log in.
- Use headphones. Echo cancellation doesn't always work, and it is distracting to a speaker to hear their voice echoed.
- Test your microphone beforehand to ensure that the recorded audio is clear. Some builtin microphones produce speech that is difficult to understand, and it is fatiguing for listeners to try to decipher noisy audio.
- Mute your microphone when not speaking to avoid inadvertent noise that may distract others.
- Turn on your camera. Nonverbal communication is important.
- Engage in the discussion. Ask questions; ask followup questions; acknowledge responses.
- Position any light source in front of you and behind the camera to best illuminate your face.
- Use a wired network connection if possible. Wireless connections may be less reliable.
- Plug laptops or mobile devices into wall power – battery use can adversely affect video quality.
- Dress appropriately.

Video DON'Ts

- Post zoom links publicly, on social media, etc. Bad actors may join the meeting and post distracting or inappropriate material.
- Post offtopic messages in the chat. It is distracting to others.
- Share private windows such as personal email.

5.4.4 Privacy and FERPA

The university and instructor value students' rights to privacy, and this course must specifically comply with the Family Educational Rights and Privacy Act (FERPA). To support FERPA compliance, please mind the following:

- Include a cover sheet on all work submitted on physical paper. The cover sheet must have the student's name and no answers or other work. Electronic submissions do not need a cover sheet.
- Use Canvas for electronic communication containing specifics about grades. Canvas is the system chosen by the university to manage students' grades.
- Do not disclose the private information (e.g. grades) of other students.

5.4.5 Collaboration Policy

- Worksheets may be completed in groups. You are encouraged to discuss worksheet exercises with others in the class.
- Homeworks must be an individual effort. You may not copy or share solutions. However, per the CS collaboration policy above, you may consult others in the class under the "empty hands" requirement.
- Projects may be completed with your project group. Per the CS collaboration policy above, you may consult other groups in the class under the "empty hands" requirement. Copying code will be considered academic misconduct.
- Exams must be an individual effort. Copying solutions or consulting others on an exam will be considered academic misconduct.

6 FAQ

- **Q:** Is the textbook "required?"
  A: Most students will need to study the textbook to learn the topics in this course. In fact, many would also benefit from studying the alternate textbooks as well.
- **Q:** When/where are office hours?
  A: The instructor will post office hours on Piazza the first or second week of the semester (it takes us some time to rearrange meeting schedules each semester). Instructor office hours are in the instructor's office (BB280H) and/or on Zoom. TA office hours will be posted and/or on Zoom.
- **Q:** What's on the exam?
  A: Exam questions will be similar to homework assignments and in-class activities. Exams will focus on evaluating understanding, application, and synthesis of the course topics (i.e., the upper levels of Bloom's taxonomy). Questions will not focus on memorization, but one must know the key definitions and concepts to apply them. For the midterm, all topics covered up to the exam may be included. The final will be cumulative but will focus on topics covered after the midterm. The instructor will post a specific list of topics after preparing each exam, typically about a week before the exam date (in past semesters, the topic list included 80-90% of the lecture material).
- **Q:** When is the midterm?
  A: Please see the tentative schedule in this document for an approximate time. The instructor will announce firm details about the midterm closer to the date and will post the details on Piazza.
- **Q:** When/where is the final exam?
  A: The registrar schedules all final exams. Please see the registrar's website.
- **Q:** Debug my code for me.
  A: The instructor and TAs are here to help you with projects but typically cannot do the job of debugging for you. Plus, learning how to debug your own code is an absolutely necessary skill.
- **Q:** What's my grade?
  A: The exact answer is unknowable until the end of the semester. For an approximate answer, see section 3 and compare your scores to the class distribution, which will typically be posted on Piazza for major assignments. Historical score distributions and grade cutoffs are listed in Appendix B.

- **Q:** Can I have an extension on an assignment?
  A: In case of extenuating circumansances (medical issue, personal emergency, etc.), of course; please contact the instructor/TA. In exceptional cases, it may be appropriate to extend a deadline for the entire class, but such extensions may also be unfair to students who completed work by the original deadline. (see "Fairness" in section 3). If you think there is reason to extend a deadline for the class, please make the request well in advance of the deadline.

- **Q:** How can I improve my grade?
  A: Participate in lecture, come to office hours, study, ask questions, and start assignments early. (see "Fairness" and "Grading Corrections" in section 3)

- **Q:** Why are exam scores so "low"? (Half the class "failed!")
  A: This course does not use an 90/80/70-percent scale. Such a scale is (1) arbitrary and (2) poorly-aligned with open-ended and challenging nature of upper-level and graduate courses such as this one. In particular, rubrics (see Appendix A) for problem solving and proofs (both of which are a focus in this course) do not align well with a 90/80/70 scale, so lower scores do not necessarily indicate "failure." Rather, this course is graded on a curve based on the statistical distribution of course scores AND observed student effort.

- **Q:** Why does this course grade on curve?
  A: While there are arguments for and against curved grading, certain factors in this course support grading on a curve. Overall, curving supports *robust* determination of letter grades that are *fair* and *consistent*. Specifically:
  - The open-ended and challenging nature of assessments in an upper-level course results in a wider distribution of scores than low-level courses that evaluate more limited outcomes (i.e., the lower levels of Bloom's taxonomy). Curving accommodates this wider distribution to produce grades that reflect learning outcomes.
  - Average scores change slightly over different terms, e.g., based on variations in difficulty of exam questions. Curved grading ensures that letter grades remain consistent.
  - Many instructors employ ad-hoc curving if letter grade distributions don't match their intent. Instead, the systematic curved grading used in this course determines letter grades based on score statistics, eliminating ad-hoc decisions about what, when, and how to curve and thus providing better consistency and fairness in the final letter grades.

- **Q:** Why does this course use...
  - **Q:** ...Microsoft OneDrive?
    A: MSOneDrive is the file storage system that ITS has chosen. RClone supports MSOneDrive, and the result is adequately usable.
  - **Q:** ...Git and Github?
    A: In previous years, when students submitted tarballs on Canvas, they often struggled to share code with each other, and groups occasionally submitted incorrect versions of their project (resulting in much lower scores than the group expected!). Git (and Github) are critical tools to collaborate on code and to reduce the chance of submitting an unintended version. Moreover, Git is pervasive in professional software development.
  - **Q:** ...OCaml
    A: OCaml is a functional programming that is common in industry and offers a syntax similar to the textbook Lambda Calculus (the mathematical foundation of programming). Thus, OCaml aligns well with the *theory & practice* philosophy of Mines.

  - **Q:** . . . Linux?
    A: Primarily, the programming tools used in the course are best supported on Linux. Secondarily, the instructor is unable to provide support for non-Linux systems (limited support for unix-like systems such as Mac OSX may be possible). Additionally, Linux proficiency is vital for computing professionals, given the pervasive use of Linux in mobile devices, cloud computing, high performance computing, robotics, etc.

- **Q:** How does the instructor prefer to be addressed?
  A: Preferred: *FIRST-NAME*, {Dr., Prof.} *LAST-NAME*
  Not preferred: {Dr., Prof.} *FIRST-NAME*, Mr. {*FIRST-NAME*, *LAST-NAME*}, "Hey!"

7 COVID Addendum

7.1 Course Procedures

The initial plans are to run this course in-person and at full capacity. However, at least some parts of the course (e.g., office hours) may be remote.

Be ready for change We cannot fully predict the course of the virus, but we can prepare. While current plans are for an in-person course, we have successfully run this course remotely in the past. We will follow the data and guidance of medical experts to prioritize both your safety and education.

Required Technology

- Microphone. Please check that audio quality is adequate.
- Headphones
- Webcam
- Zoom
- Media player supporting H.264 and AAC. VLC generally works well.

Please see also subsubsection 5.4.3.

Project Collaboration Please use the full array of software collaboration tools to support collaboration on group projects: git, github, email, text chat, video chat, etc.

Possibility to Alternate In-Class Attendance If necessary, we will divide the class in halves to attend on alternating days. Smaller groups reduce in-person contact and may provide a better atmosphere for discussion. Students should watch any posted lecture videos prior to attendance, so that reduced in-class time can focus on questions and exercises.

In case of illness If you or a family member become ill or face specific COVID-related challenges, please contact the instructor. We will make whatever accommodations are appropriate to deal with these extenuating circumstances.

If the instructor becomes ill, every course in the department, including this one, has another designated faculty member to substitute.

In case of instructor quarantine If the instructor must quarantine, but still feels well enough to teach, we will hold class remotely on Zoom. The instructor will attempt to provide as much advance notification of such a change as possible, but sometimes significant advance notice is not possible. Please regularly monitor your email and the course Piazza for such changes. If you in the classroom and the instructor is not, check your email / Piazza!

7.2 Oredigger Promise

(From Oredigger Promise: Our Climb Continues)

We made it through the 2020-21 school year together, on campus and in person, and we can do it again this year.

But just as global data trends show us the pandemic isn't over, we also know it will take a shared commitment from us all to safely navigate the year ahead.

As our understanding of the virus increases and variants take center stage, our commitment to protecting our community needs to evolve. This year's Oredigger Promise reflects this evolution as well as our continued need to climb together and protect our classmates and colleagues, our families and neighbors.

I Will:

- *Monitor my health daily and check for COVID-19 symptoms* I will stay home if I am experiencing symptoms of COVID-19 – even if I feel well enough to come to campus and even if I'm vaccinated. I will get tested for COVID-19 before returning to campus life. I will report symptoms of or exposure to COVID-19 to Mines.
- *Isolate and quarantine as directed* Isolation is for those who test positive for COVID-19 and quarantine is for those who have had close-contact exposure to someone with COVID-19.
- *Wear an appropriate face covering* over my mouth and nose when inside classrooms, teaching labs, computer labs and other campus locations designated at entrances. I will also be supportive if others choose to wear face masks in other spaces around campus.
- *Wash my hands frequently* using soap and water or hand sanitizer.
- *Participate in contact tracing and testing* to preserve the wellness of the community.
- *Be gracious and attentive* when others provide safety reminders and when I notice a fellow Oredigger who may be struggling.

A Proof Rubric

This course includes proofs. We will use the following rubric to grade proofs.

Logic and Reasoning: 50%

- 0%: Totally wrong, all steps in the argument are illogical.
- 10%: At least one step is valid, but the argument is mostly wrong.
- 20%: Partially valid argument, but significant errors in the steps or conclusions.
- 30%: Overall argument is generally valid, but major errors present.

- 40%: Conclusion is valid, but minor errors present.
- 50%: All steps are correct and the argument is valid.

Communication – Terminology and Notation: 20%

- 0%: Major errors in terminology indicating a lack of understanding.
- 10%: Minor errors in terminology or imprecise/informal language.
- 20%: Correct, formal, and precise use of terminology.

Communication – Structure: 20%

- 0%: Difficult or impossible to follow.
- 10%: Follows basic structure for the type of proof.
- 20%: Flows well, suitable concise, and easy to read.

Communication – Grammar: 10%

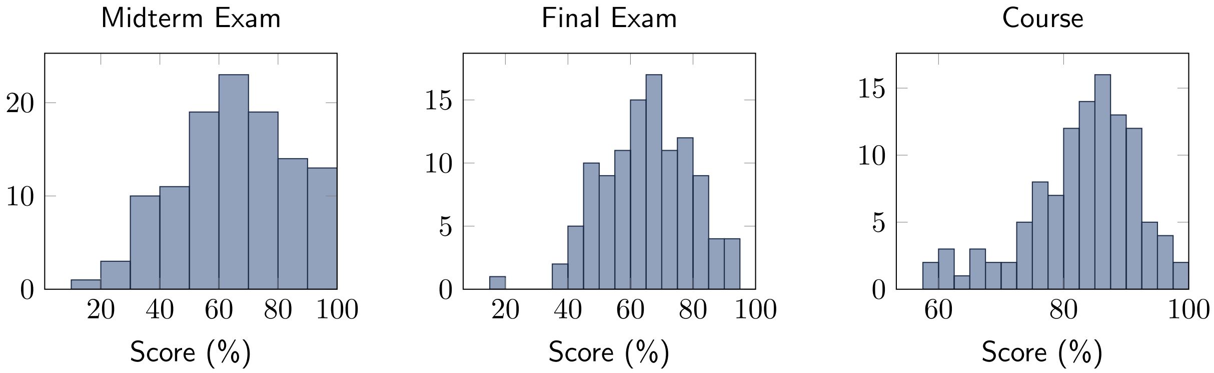
- 0%: Grammar problems impede understanding.
- 10%: Sufficiently correct grammar. No issues that affect understanding.

B Historical Score Distributions

Below are score distributions from prior terms. Letter grades are based on the Course Score, which is the weighted average as described in section 3 (weights may vary between terms).

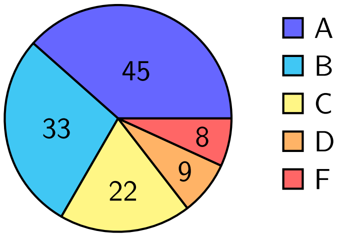
Spring 2022

Note: all sections returned to fully in-person classes, and students had access to pre-recorded lecture videos, with additional discussion and activities during class meetings. However, due to continuing COVID, many students missed class meetings for isolation or quarantine.



	Midterm	Final	Course
Mean	65.4	65.6	82.9
Median	66	66.0	84.1
Std. Dev.	19.0	14.2	8.9

A	$86.0 \leq s$
B	$78.0 \leq s < 86.0$
C	$70.0 \leq s < 78.0$
D	$63.0 \leq s < 70.0$



Spring 2021

Note: due to continuing COVID, one section of the course with 28 students was taught face to face and one section with 52 students was taught remotely. Both sections had access to pre-recorded videos, with discussion during class meetings.

	Midterm	Final	Course
Mean	59.5	53.2	78.1
Median	61	50.75	80.5
Std. Dev.	18.5	22.2	10.1

A	$85.0 \leq s$
B	$76.5 \leq s < 85.0$
C	$66.0 \leq s < 76.5$
D	$55.0 \leq s < 66.0$

Spring 2020

Note: the university switched to emergency remote learning partway through the Spring 2020 semester. Letter grades in this course were all increased by roughly half a letter to compensate for the increased difficulty of this emergency switch.

	Midterm	**Final**	**Course**
Mean	51	70.5	74.1
Median	53	76	76.5
Std. Dev.	18.5	23.6	14.6

A	$80.0 \leq s$
B	$70.0 \leq s < 80.0$
C	$60.0 \leq s < 70.0$
D	$54.0 \leq s < 60.0$