COLORADOSCHOOLOFMINES
EARTH • ENERGY • ENVIRONMENT

CSCI-561: Theory of Computation (Theory)

Term: Fall 2022
Hours: 3 cr
Class Meetings: 2:00-3:15pm, Tue/Thu, MZ 126
Instructor: Dr. Neil T. Dantam
Office Location: BB280H
TAs: Benjamin Wagley and Noah Fields

Contents

1 Overview and Outcomes

1.1 Course Description

Are there "laws of physics" for computing? Are there fundamental limits to what computers can do—and thus things computers cannot do? If so, what makes computational problems harder or easier, solvable or unsolvable? And when faced with a new computational problem, how can we determine its difficulty or solvability?

In this course, we will address such questions about the fundamental capabilities and limits of computation. In particular, we will answer the following:

- *What is a computer?*
  We will study different models of computation.
- *What can we compute?*
  We will define problems that are solvable/unsolvable using different models of computation.
- *How well can we compute?*
  We will analyze the performance capabilities and limits for various computational models and problems.

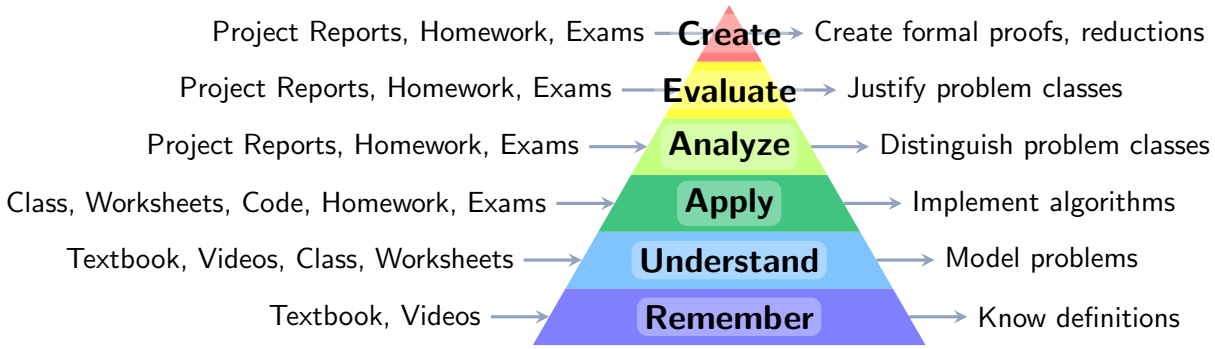


Figure 1: Overview of course activities and outcomes through Bloom's Taxonomy

At many universities, courses on the *Theory of Computation* are purely theoretical, in essence, math classes. Here at Mines, we aim to blend theoretical rigor and practical application. Thus, in this course, we will both study fundamental results of computational theory and reduce theory to practice though projects that implement and apply key algorithms of theoretical computer science.

1.2 Learning Outcomes

While this course focuses on a particular set of foundational topics in computer science, the broader aim is to help you develop and grow the formal thinking, analysis, and communication skills that will serve you throughout your career in computer science. We will study topics in language theory and complexity theory. In language theory, we will study representations such as finite automata, regular expressions, context-free grammars, pushdown automata, and Turing machines and properties such as (non)determinism, decision properties, and closures properties. In complexity theory, we will study time complexity, NP-completeness, and space complexity.

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

- Communicate effectively using mathematically precise terminology, particularly about topics in language and complexity theory.
- Implement language theory algorithms for common decision and closure properties.
- Model and solve computational problems using formal language representations and algorithms. We will study classic applications of formal languages to text processing as well as additional applications such as discrete event systems and automated planning.
- Evaluate potentially effective (or ineffective) approaches for new problems based on language and complexity class.
- Formally prove capabilities and limits, such as decidability and complexity of new problems, based on computational classes and using methods such as construction, contradiction, pumping lemmas, and reductions.
- Organize and manage medium-size (multi-week) group projects involving programming, analysis, and written communication (reports).

2 General Course Information

Prerequisites The official prerequisite is CSCI 406: Algorithms. Generally, prerequisites are:

- Comfort with discrete math / set theory, recursion.
- Moderate programming experience is required. Prior experience with functional programming is recommended (e.g., CSCI 400).
- Moderate experience with proofs.

Textbooks and References

- **Primary Textbook** (main reference for the course) Michael Sipser. *Introduction to the Theory of Computation*. ISBN 978-0357670583.
- **Supplemental Textbooks** (reference for some advanced topics)
  - John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. ISBN : 978-0321455369
  - Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman. *Compilers: Principles, Techniques & Tools*. ISBN 978-0321486813.
- **Lisp Programming References**
  - Peter Siebel. *Practical Common Lisp*. ISBN 978-1430242901. `http://www.gigamonkeys.com/book/`
  - *Common Lisp HyperSpec*. `http://www.lispworks.com/documentation/HyperSpec/Front/`
  - Paul Graham. *ANSI Common Lisp*. ISBN 978-0133708752.
- **Supplemental Materials** (additional videos, notes, and practice exercises)
  - MIT 6-045J: Automata, Computability, And Complexity, Spring 2011 (Includes notes and problem sets)
  - MIT 18.404J: Theory of Computation, Fall 2020 (Includes videos, notes, problem sets, and exams)

Studying the textbook(s) is a valuable (and expected) learning activity. Homeworks will not be assigned out of the textbooks, so older editions are usually sufficient and can often be purchased at a much lower price than the newest edition. The instructor originally prepared the content for this course from the 1st edition of Hopcroft, 2nd edition of Sipser, and 2nd edition of Aho.

Office Hours

- Instructor office hours will be both face-to-face (BB280H) and on Zoom (see Piazza for the link).
  - TBD
- TA Office Hours locations are TBD.
  - 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 Piazza for announcements and assignments, 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 worksheet 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. Please limit specific discussion of grades to Canvas.
- **Concerns/suggestions about course procedures** Email the instructor or TAs about the issue and/or to schedule a meeting to discuss the issue.

3 Assessments and Grading

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

Worksheets and Participation	10%	(w)
Quizzes	10%	(q)
Homeworks	10%	(h)
Projects	30%	(p)
Midterm Exam	15%	(m)
Final Exam	25%	(f)

$$\overbrace{s}^{\text{Course Score}} = .1\left(\frac{\mathtt{w}_{\text{recv.}}}{\mathtt{w}_{\text{poss.}}}\right) + .1\left(\frac{\mathtt{q}_{\text{recv.}}}{\mathtt{q}_{\text{poss.}}}\right) + .1\left(\frac{\mathtt{h}_{\text{recv.}}}{\mathtt{h}_{\text{poss.}}}\right) + .30\left(\frac{\mathtt{p}_{\text{recv.}}}{\mathtt{p}_{\text{poss.}}}\right) + .15\left(\frac{\mathtt{m}_{\text{recv.}}}{\mathtt{m}_{\text{poss.}}}\right) + .25\left(\frac{\mathtt{f}_{\text{recv.}}}{\mathtt{f}_{\text{poss.}}}\right)$$

Worksheets and 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.

Midterm Exam A midterm exam will take place around the middle of the semester.

Final Exam A cumulative exam will take place during finals week.

Homeworks There will be several homeworks and exercises.

Projects There will be a warmup plus two projects on applications of CS theory. The amount of code you will need to write is fairly small (a few hundred lines at most). However, you will need to think carefully about the relevant theory, math, and algorithms. Thus, **it is critical that you start projects early** so you have sufficient time to think through the required implementation and application (and ask questions if you get stuck).

Project 0 Warm-up project on programming environment and mathematical preliminaries.

Project 1 Finite Automata and Regular expressions.

Project 2 Propositional Logic and Boolean Satisfiability.

Letter Grades 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. Assuming consistent, normal distribution of scores, the A/B cutoff will be approximately at the median score, and scores more than one standard deviation below the average may receive less than a B. However, skewed student effort (including absenteeism) or score distributions may result in correspondingly skewed letter distributions.

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 C-. 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 C-. This requirement is to encourage attendance and participation in the class meetings.

3. To receive a passing grade, the weighted average of the midterm and final exam score must exceed the lesser of 35% and two standard deviations below the class average. This requirement is to address the potential "free-riding" on project group members.

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.

The deadline to request a grade correction is two weeks from the date an assignment is returned or the grade is posted.

Projects Expectations and Grading

- Projects will include a coding portion and a report portion.
- 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.
- Report grading will evaluate your overall understanding for the project area.
- Projects will include a peer evaluation of group members. In cases where the is evidence of unequal contributions among group members—based on peer evaluations, git commit history, student interviews, etc.—the instructor may adjust project scores to better reflect individual contributions.

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.

- Include page numbers and total page count in written reports to ensure pages are properly ordered and no pages are overlooked.
- 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

(updated 2022-11-12)

W	Date	Topic	Files	References (Sipser/Hopcroft/Aho)
1	Aug 23	00: Introduction to Theory of Computation		Syllabus
	Aug 25	01: Math Review		S 0, H 1
2	Aug 30	02: Symbolic Computation and Common Lisp		Lisp References
	Sept 1	03: Functional Programming in Lisp		Lisp References
3	Sept 6	04: Finite Automata		S 1.1-1.2, H 2.1-2.3, A 3.6-3.7
	Sept 8	05: Subset Construction		S 1.1-1.2, H 2.3, 2.5, A 3.6-3.7
4	Sept 13	Career Day (no class)		
	Sept 15	06: Regular Expressions		S 1.2-1.3, H 3, A 3.7
5	Sept 20	07: McNaughton-Yamada-Thompson Algorithm		S 1.2-1.3, H 3, A 3.7
	Sept 22	08: Regular Decision Properties		S 4.1, H 4.3
6	Sept 27	09: Pumping Lemma		S 1.4, H 4.1
	Sept 29	10: Regular Closure Properties		H 4.2
7	Oct 4	11: Finite Automata Minimization		H 4.4, A 3.9
	Oct 6	Catchup and/or Lab Day for Project 1		
8	Oct 11	Midterm Review (tentative)		
	Oct 13	Midterm (tentative)		
9	Oct 18	Fall break (no class)		
	Oct 20	12: Application: Discrete Event Systems		Cassandras & Lafortune
10	Oct 25	13: Boolean Satisfiability		S 7.1, H 10.2, Russell & Norvig 7.6
	Oct 27	14: Application: SATPlan		LaValle 2
11	Nov 1	15: Grammars		S 2.1, H 4.1-4.2, A 4.2
	Nov 3	Catchup and/or Lab Day for Project 2		
12	Nov 8	16: Pushdown Automata		S 2.2, H 6
	Nov 10	17: Pumping Lemma for Context-Free Languages		S 2.3, H 7
13	Nov 15	18: Context-Free Languages		H 7
	Nov 17	19: Application: Parsing		H 5.2-5.3, A 4.4
14	Nov 22	20: Turing Machines		S 3, H 8
	Nov 24	Thanksgiving Break (no class)		
15	Nov 29	21: Decidability		S 4-5, H 9
	Dec 1	22: Complexity		S 7, H 10
16	Dec 6	TBD / Catchup / Review		
	Dec 8	Review Day		
17	Dec 13	Finals Week (see the Registrar's exam schedule for time and location)		
	Dec 15	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.

- **Prepare before class:**
  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.
- **Participate during class:**
  1. Ask questions about the video and worksheet exercises.
  2. Discuss the lecture topics.
  3. Take a quiz.
  4. Participate in additional activities such as coding exercises and group proofs. **Many in-class activities will closely resemble exam questions.**

Preparation *before class and* **participation** *during class are both expected and required parts of this course.*

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 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 and quizzes must be an individual effort. Copying solutions or consulting others on an exam or quiz will be considered academic misconduct.

5.4.4 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.

5.4.5 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.

A FAQ

- **Q:** Is the textbook "required?"
  A: Studying the textbook is an expected learning activity for this course.
- **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:** The syllabus says that Linux is the supported platform for course projects, but can I use macOS?
  A: MacOS does provide some unix-like features, but it also does some things differently. If you encounter a mac-specific problem, the instructor and TAs won't be able help. In other words, it might work just fine, but if it doesn't, you'll have to figure it out yourself.
- **Q:** Which Linux distribution and version is best?
  A: Debian and Ubuntu (which is based on Debian) are good choices for distributions and have easy and robust package managers. Debian "Stable" and Ubuntu "Long-term Support (LTS)" versions will have the most thorough testing and least likelihood of encountering problems.

- **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 and grade statistics are listed in Appendix C.
- **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. Score changes after-the-fact are not possible, i.e., do not ask about grade improvements after the semester has ended but instead prepare throughout the semester so you can best achieve the course's learning outcomes. See also: "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 B) 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 a grad-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:** . . . Common Lisp?
    A: Many algorithms in this course are more naturally expressed in the functional programming style, recursion and induction (typical of functional programs) are necessary to understand the algorithms and proofs we will cover, and symbolic manipulation is a key aspect of many of these algorithms. Lisp is a good language for functional programming and an excellent language (arguably the best) for symbolic manipulation, so understanding and using Lisp will help you understand and implement the algorithms we will cover. Moreover, the representation of programs in Lisp offers fundamental insight into the meaning of programming and computing. As described by Alan Kay (inventor of OOP), Lisp contains the "Maxwell's equations of programming." – a wonderful description of the type of insight we hope to gain in this course.
  - **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—though not explicitly a learning outcome of this course—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!"

B 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.

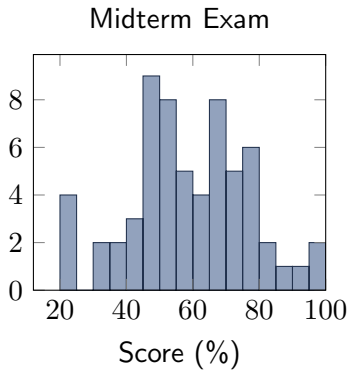
Communication – Grammar: 10%

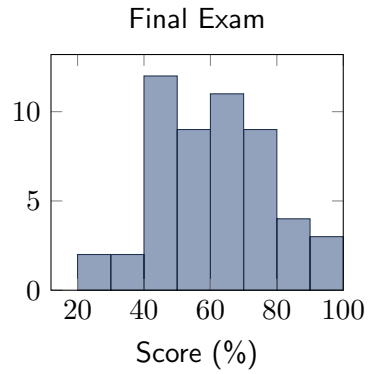
- 0%: Grammar problems impede understanding.
- 10%: Sufficiently correct grammar. No issues that affect understanding.

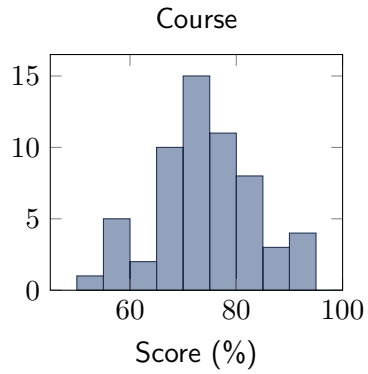
C Historical Score and Grade Distributions

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).

Fall 2021

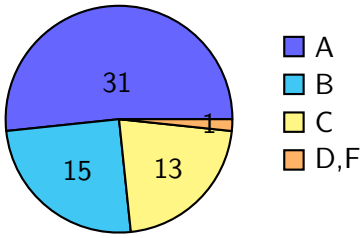




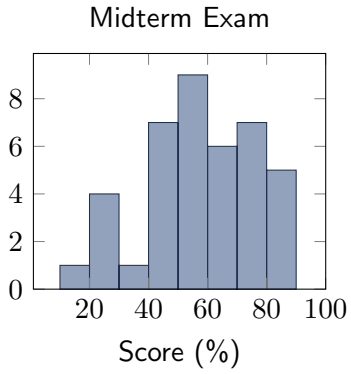


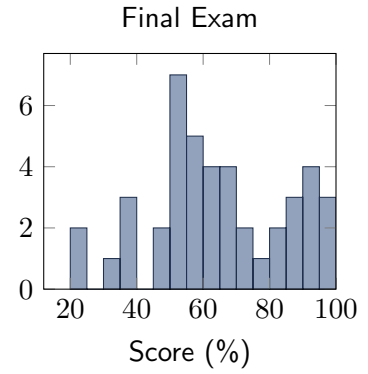
	Midterm	Final	Course
Mean	60.8	61.1	73.9
Median	60	59.5	73.5
Std. Dev.	16.6	16.2	9.48

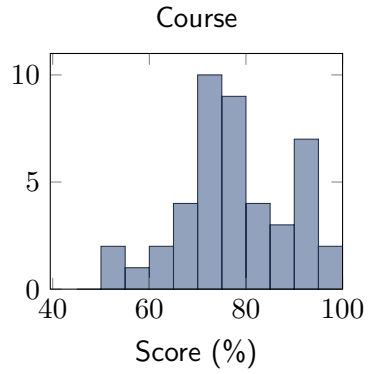
A	$73 \leq s$	31
B	$68 \leq s < 73$	15
C	$52 \leq s < 68$	13
D,F	$s < 52$	1
	Total	60



Fall 2020

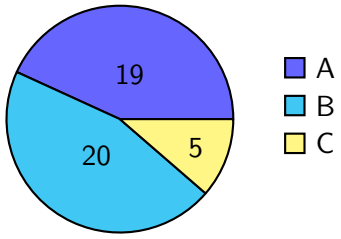




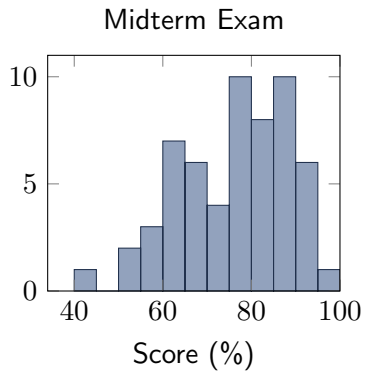


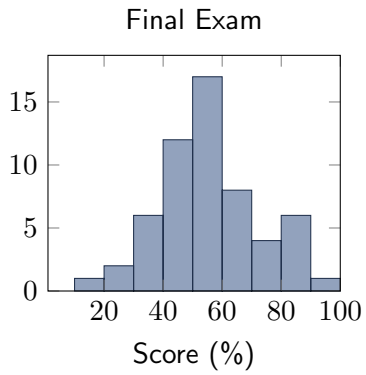
	Midterm	Final	Course
Mean	64.1	64.8	77.8
Median	61.5	61.5	75.7
Std. Dev.	21.6	20.3	10.9

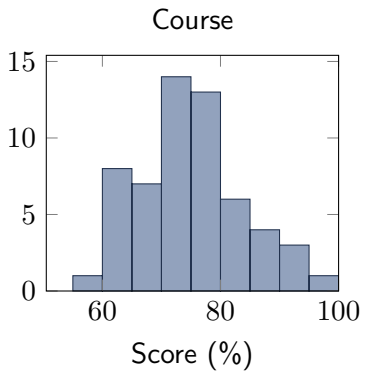
A	$77 \leq s$	19
B	$65 \leq s < 77$	20
C	$50 \leq s < 65$	5
D,F	$s < 50$	0
	Total	44



Fall 2019







	Midterm	Final	Course
Mean	77.6	55.9	74.3
Median	79.0	53	74.4
Std. Dev.	11.7	16.9	8.92

A	$74 \leq s$	31
B	$61 \leq s < 74$	24
C	$50 \leq s < 61$	1
D,F	$s < 50$	0
	Total	56