COLORADOSCHOOLOFMINES
EARTH • ENERGY • ENVIRONMENT

CSCI-561: Theory of Computation (Theory)

Term: Fall 2021
Instructor: Dr. Neil T. Dantam

1 Overview and Outcomes

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 and 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.

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

Remember: Know definitions of conventional objects in language and automata theory.
Example: Define a context-free grammar.

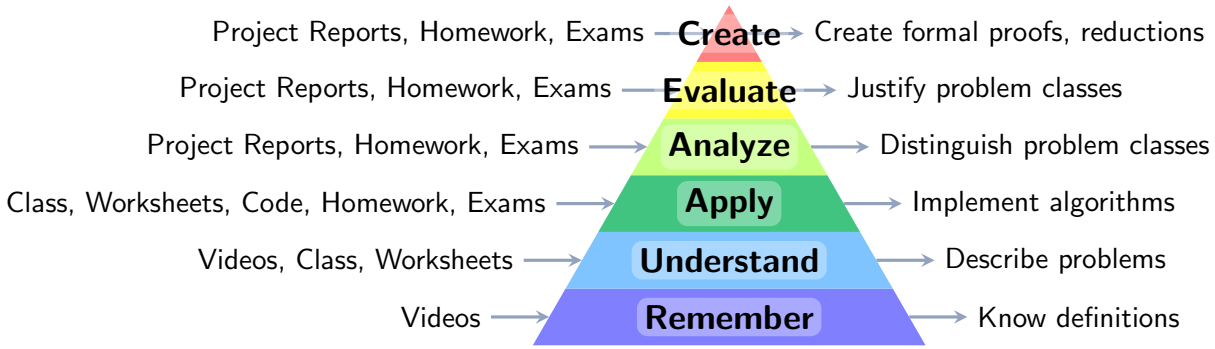

Figure 1: Bloom's Taxonomy of Learning Activities and Outcomes

Understand: Describe computational problems using formal languages.
Example: Write a regular expression to find email addresses

Apply: Implement existing language theory algorithms
Example: Write code to convert a regular expressions to a finite automaton.

Analyze: Distinguish suitable computational classes for new problems.
Example: Could we model some X as a regular language and/or solve via Boolean Satisfiability?

Evaluate: Justify the suitability of various computational classes for new problems.
Example: Why should we use context-free grammar vs. regular expressions to parse a particular file format?

Create: Develop proofs and reductions (algorithmic transformations) to characterize the required computation and/or solve a new problem.
Example: Create a formal proof that a file format cannot be parsed with regular expressions.

2 General Course Information

Instructor: Dr. Neil T. Dantam
TA: Justin McGowen

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*.
- **Alternate Textbooks** (reference some advanced topics)
  - John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*.
  - Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman. *Compilers: Principles, Techniques & Tools*.
- **Lisp References**
  - Peter Siebel. *Practical Common Lisp*. http://www.gigamonkeys.com/book/
  - *Common Lisp HyperSpec*. http://www.lispworks.com/documentation/HyperSpec/Front/
  - Paul Graham. *ANSI Common Lisp*. http://www.paulgraham.com/acl.html

Generally, older editions of the above textbooks are 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/Google Calendar for the link).
  - Tuesday, 2-3pm
  - Thursday, 11:30am-12:30pm
- TA Office Hours are remote. (see Piazza/Google Calendar for the link).
  - Monday, 3-4pm
  - Wednesday: noon-12:50pm

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
- Google Calendar: Lecture and Office Hour Schedule

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. The instructor and TAs can provide only very limited technical support if you attempt to use a non-Linux platform.

Many class activities will require an electronic device (smartphone, tablet, or laptop). Some class activities specifically require a Linux laptop (running Linux directly, in a virtual machine, SSHing to Isengard, etc.). If bringing an electronic device and/or Linux laptop to class presents difficulties for you, please contact the instructor about possible arrangements or alternatives.

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 Grading and Evaluation

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

Class Participation	10% (c)
Homeworks	10% (h)
Projects	35% (p)
Midterm Exam	18% (m)
Final Exam	27% (f)

$$\texttt{score} = .1\left(\frac{\texttt{c}_{\text{recv.}}}{\texttt{c}_{\text{poss.}}}\right) + .1\left(\frac{\texttt{h}_{\text{recv.}}}{\texttt{h}_{\text{poss.}}}\right) + .35\left(\frac{\texttt{p}_{\text{recv.}}}{\texttt{p}_{\text{poss.}}}\right) + .18\left(\frac{\texttt{m}_{\text{recv.}}}{\texttt{m}_{\text{poss.}}}\right) + .27\left(\frac{\texttt{f}_{\text{recv.}}}{\texttt{f}_{\text{poss.}}}\right)$$

Class Participation Most lectures will have a worksheet to practice the material. After the lecture is complete (i.e., we finish the set of slides corresponding to the worksheet), scan or photograph the worksheet and submit it on Canvas. Your participation grade for the worksheets will be based on making an honest effort. During lecture, we may also have additional activities that will count towards the participation grade.

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 or score distributions may result in correspondingly skewed letter distributions.

Additional Requirements to Pass In addition to the scoring and letter grade determination discussed above, the following requirements must be met to pass this course:

1. At least 80% of participation assignments must be attempted and submitted on time.
2. The greater of the final exam score and the weighted average of the midterm and final exam score must exceed the lesser of 35% and two standard deviations below the corresponding class average score.

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

- Projects will include a coding portion and a report portion.
- Code will be graded objectively. 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.

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 reports, homeworks, 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 2021-12-10)

W	Date	Topic	Files
1	Aug 24	00: Introduction to Theory of Computation	
	Aug 26	01: Math Review	
2	Aug 31	02: Symbolic Computation and Common Lisp	
	Sept 2	03: Functional Programming in Lisp	
3	Sept 7	04: Finite Automata	
	Sept 9	05: Subset Construction	
4	Sept 14	Career Day (no class)	
	Sept 16	06: Regular Expressions	
5	Sept 21	07: McNaughton-Yamada-Thompson Algorithm	
	Sept 23	08: Regular Decision Properties	
6	Sept 28	09: Pumping Lemma	
	Sept 30	10: Regular Closure Properties	
7	Oct 5	11: Finite Automata Minimization	
	Oct 7	Catchup and/or Lab Day for Project 1	
8	Oct 12	Midterm Review (tentative)	
	Oct 14	Midterm (tentative)	
9	Oct 19	Fall break (no class)	
	Oct 21	12: Application: Discrete Event Systems	
10	Oct 26	13: Boolean Satisfiability	
	Oct 28	14: Application: SATPlan	
11	Nov 2	15: Grammars	
	Nov 4	Catchup and/or Lab Day for Project 2	
12	Nov 9	16: Pushdown Automata	
	Nov 11	17: Pumping Lemma for Context-Free Languages	
13	Nov 16	18: Context-Free Languages	
	Nov 18	19: Application: Parsing	
14	Nov 23	20: Turing Machines	
	Nov 25	Thanksgiving Break (no class)	
15	Nov 30	21: Decidability	
	Dec 2	22: Complexity	
16	Dec 7	TBD / Catchup / Review	
	Dec 9	Final Review	
17	Dec 14	Finals Week	
	Dec 16	Finals Week	

5 Policies

5.1 Mines Policies and Resources

Mines Policies and Resources

5.2 CS Collaboration Policies

CS Collaboration Policies

5.3 Course Policies

5.3.1 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: Students will watch a lecture video and attempt a set of practice exercises (worksheets) corresponding to examples shown in the video.

During Class: We will answer questions about the video and worksheet exercises, and we will conduct additional activities such as discussion, group proofs, and coding exercises.

After Class: Students will submit the completed worksheet exercises and activities on Canvas to receive participation credit.

5.3.2 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., 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.

5.3.3 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.

5.3.4 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.

5.3.5 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.

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 the homework assignments and will focus on evaluating understanding, application, and synthesis of the course topics (i.e., the upper levels of Bloom's taxonomy), including specifically writing proofs. 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 Canvas.

- **Q:** When/where is the final exam?
  A: The registrar schedules all final exams. Please see the registrar's website.

- **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 Canvas for major assignments.

- **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 other cases, sometimes it may be appropriate to extend a deadline for the entire class (see "Fairness" in section 3).

- **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. Instead, this course is graded on a curve based on the statistical distribution of course scores.

- **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 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 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 5.3.5.

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.

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.