Introduction

Discrete Structures

This course is an introduction to the mathematical building blocks—formal logic and proof techniques, combinatorics, recursion, and graph theory—needed for problem solving in computer science.

This is a math class, but it is a different flavor of mathematics than you may have seen in an algebra/trigonometry or calculus course. Whereas the beauty in calculus comes from unifying the physical with the algebraic and geometric, the beauty in a discrete course lies in the structures and patterns that emerge from simple axioms. For example:

- **Recursion** is a common pattern in computer science that includes the generation of fractals and other self-similar structures, as well as the design of algorithms. We will look at what recursion is, taking a formal view that helps us to understand how one constructs recursive structures and algorithms.
- **Graphs** have a simple set-theoretic definition and can model any relational structure. Even graphs of modest size can exhibit a variety of interesting behavior and patterns.
Sets underlie much of what we do mathematically and as computer scientists. They are the building blocks on which structures such as graphs, functions, sequences, and series are built. Logic is the formal process of reasoning. We will look at this from the ground up, considering both symbolic logic of the form often done by machines, and the practice of writing rigorous proofs. The course also introduces combinatorial proofs, proofs by induction, and other proof techniques—ways of sharpening one's thinking about why a mathematical fact is true.

Syllabus purpose

One purpose of this syllabus is to give you an idea of what concepts will be covered in this course, how I will assess your learning, and what you will be graded on. Zooming out from that goal, the syllabus is how I communicate to you what the purpose of the course is, why it is structured the way it is, what the goals of the course are, and what the intended learning outcomes are. Even more broadly than that, the syllabus is where classroom norms are documented: expectations regarding how we all engage and interact in the classroom. More broadly still, I have listed some campus resources that may be helpful.

About me

My name is Daniel Frishberg. I use he/him pronouns. I prefer to have students call me Daniel, but Professor Frishberg or Dr. Frishberg is fine too.

My web page: https://users.csc.calpoly.edu/~dfrishbe/

Education:

- I finished my PhD in Spring 2023 in theoretical computer science at the University of California, Irvine (UCI), advised by Professor David Eppstein.
- I have a Master’s (MS) in Computer Science from UCI (awarded in 2020).
- My Bachelor’s degree was in Computer Science from Oberlin College in 2007.

Research interests:

- random sampling algorithms via Markov chains: theoretical analysis, with applications to graphical models in machine learning
- graph theory and algorithms, including network flows, puzzle games, and directed acyclic graphs (applications include computer network infrastructure and project planning)
- computational geometry, including graph drawings and point set triangulations (applications include computational biology)

Industry experience:

- I worked as a web developer for 11 years.
- I built Python testing frameworks, DevOps tools, and JavaScript UI libraries for the National Institutes of Health (NIH).
I did Google software and research internships (working on network infrastructure) while in graduate school.

Learning outcomes

Something you may have seen on other course syllabi is the use of student learning outcomes (SLOs). These are in some sense the entire point of the course. They capture what I as an instructor, and Cal Poly as an institution, would like you to get out of this course. These are also the skills I plan to assess your learning on, with tests and programming projects- more on that in the "Assessments and grading" section.

At the same time, this is about your education. You may have additional outcomes that are important to you. I encourage you to read these outcomes, think about what your goals are, for your education and perhaps your career, and what outcomes you may want out of this course that may or may not be in this list. I would love it if you even come to office hours and discuss some of these outcomes with me, or email me, so I know better how to design the course.

Student learning outcomes

After completing this course, students will be able to…

1. Apply formal methods of symbolic propositional and predicate logic.
2. Formulate formal logic proofs and apply logical reasoning to solve problems.
3. Determine which type of proof is best for a given problem.
4. Explain, with examples, the basic terminology of functions, relations, and sets.
5. Perform the operations associated with sets, functions, and relations.
6. Explain the asymptotic behavior of functions describing time complexity.
7. Demonstrate basic counting principles.
8. Solve recurrence equations.
9. Illustrate, by example, the basic terminology of graph theory.
10. Model problems in computer science using graphs and trees.

Course themes

Roughly speaking, there are four major themes in the course:

Theme 1: Logic and proofs, induction and recursion

Learning outcomes: After completing this course, students will be able to…

- Apply formal methods of symbolic propositional and predicate logic.
- Explain, with examples, the basic terminology of functions, relations, and sets.
- Formulate formal logic proofs and apply logical reasoning to solve problems.
- Determine which type of proof is best for a given problem.

**Content:** We take a dive into the foundations of mathematical reasoning, exploring some of the basic principles of logic and proof techniques, as well as symbolic logic. These topics have broad applications in algorithms and data structures, complexity theory, programming languages and compilers, databases, computer architecture, and AI.

**Theme 2: Functions and recursion**

**Learning outcome:** After completing this course, students will be able to...
- Explain, with examples, the basic terminology of functions, relations, and sets.
- Perform the operations associated with sets, functions, and relations.
- Solve recurrence equations.

**Content:** Like logic, sets, and proofs, functions are fundamental to virtually all of mathematics. They are also important for understanding the structure of computer programs—and for designing algorithms. Sets and functions lie at the heart of cryptography, machine learning, and probability. Recursion is essential in software engineering, algorithm design and analysis, theory of computation, compiler design, and graph theory.

**Theme 3: Counting and probability**

**Learning outcome:** After completing this course, students will be able to demonstrate basic counting principles.

**Content:** We start this unit with some natural questions that are easy to state: counting the ways to organize and choose different kinds of objects. This has applications in algorithm analysis and complexity theory, as well as security and cryptography. The questions in this unit motivate the rigor we will develop in the units on logic and sets, and the techniques in this unit lay the groundwork for graph theory and algorithm analysis later in the course.

**Theme 4: Graphs, algorithms, and asymptotic analysis**

After completing this course, students will be able to...
- Explain the asymptotic behavior of functions describing time complexity.
- Illustrate, by example, the basic terminology of graph theory.
- Model problems in computer science using graphs and trees.

**Content:** In this unit we combine the components from prior units and begin exploring graphs, asymptotic notation, and algorithm analysis.
Schedule of topics

This is a rough overview of the topics and assignments I expect we will cover in this session. However, some of the topics, assignments, and due dates may change.

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic(s) or Sub-topic(s)</th>
<th>Assessments and Assigned Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 1</td>
<td>● Course logistics&lt;br&gt; ● Applications of Discrete Structures&lt;br&gt; ● Prior knowledge survey (Google Form—see Canvas)&lt;br&gt; ● Start logic  ○ Predicates and proposition s&lt;br&gt; ○ Equivalenc e laws</td>
<td>● Intake survey on prior knowledge&lt;br&gt; ● Obtain textbook&lt;br&gt; ● Problem set 1 assigned (Logic and Proofs)</td>
</tr>
<tr>
<td>Week 2</td>
<td>● Continue logic  ○ Quantifiers&lt;br&gt; ● Set theory&lt;br&gt; ● Proof techniques  ○ Induction  ○ Proof by contradicti on</td>
<td></td>
</tr>
<tr>
<td>Week 3</td>
<td>● Finish proofs&lt;br&gt; ● Functions</td>
<td>● Problem Set 1 due</td>
</tr>
<tr>
<td>Week 4</td>
<td>● Bijections and countability&lt;br&gt; ● Sequences and series</td>
<td>● Quiz 1&lt;br&gt; ● Problem Set 2 assigned (functions, sequences and series)</td>
</tr>
<tr>
<td>Week</td>
<td>Topics</td>
<td>Assignments</td>
</tr>
<tr>
<td>------</td>
<td>------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>5</td>
<td>• Intro to counting</td>
<td>• Problem set 2 due (functions, sequences and series)</td>
</tr>
<tr>
<td></td>
<td>• Permutations and Combinations</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Product rule, sum rule</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>• Additional counting topics</td>
<td>• Quiz 2 (functions, sequences and series)</td>
</tr>
<tr>
<td></td>
<td>• Probability and counting</td>
<td>• Problem set 3 assigned (counting)</td>
</tr>
<tr>
<td>7</td>
<td>• Trees</td>
<td>• Problem set 3 due</td>
</tr>
<tr>
<td></td>
<td>• Recursive definitions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Structural induction</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>• Algorithms and complexity</td>
<td>• Quiz 3 Tuesday (counting)</td>
</tr>
<tr>
<td></td>
<td>• Asymptotic notation</td>
<td>• Problem set 4 assigned (graphs, algorithms, analysis)</td>
</tr>
<tr>
<td>9</td>
<td>• Graphs</td>
<td>• Problem set 4 due</td>
</tr>
<tr>
<td></td>
<td>• Structures in graphs</td>
<td>• Quiz 4 (graphs, algorithms, analysis)</td>
</tr>
<tr>
<td>10</td>
<td>• Finish graphs</td>
<td>• Problem set 4 due</td>
</tr>
<tr>
<td></td>
<td>• Review</td>
<td>• Quiz 4 (graphs, algorithms, analysis)</td>
</tr>
<tr>
<td>Final</td>
<td>Final exam</td>
<td>• 3pm Section: Final exam Monday March 18, 1:10pm-4:00pm</td>
</tr>
<tr>
<td>Week</td>
<td></td>
<td>• 4pm Section: Final exam Wednesday, March 20, 4:10pm-7:00pm</td>
</tr>
</tbody>
</table>

**Course textbook**

*Rosen: Discrete Math with Applications, 8th Edition*

There will be reading assignments most weeks, including ungraded, uncollected homework problems.

*Please let me know if purchasing this book presents a hardship for you.*
Graded work

Problem sets (25% of grade)
There will be roughly four problem sets, consisting of problems that are mathematical in nature.

Thus each problem set is worth roughly 6% of the final grade. However, the lowest problem set score will be dropped.

Topic Quizzes (35% of grade)
There will be four in-class topic quizzes.
- Each topic quiz on its own is worth roughly 9% of your grade.
- BUT if you don’t do great on a topic quiz, you can make it up on the final (see below).

Final Exam (40% of grade - but see below re. pre-emption)
- The final exam is cumulative.
- The final exam is worth 40% of your grade.
- However, as stated above, you can pre-empt some of the questions with the earlier topic quizzes.

Pre-emption
The final exam will include (but is not limited to) three “topic questions.” Each of these questions corresponds to material from one of the three topic exams. For each such question, I will count the maximum of either your score on that question, or your score on the corresponding topic exam, towards your final grade. In other words, if you get 95% on Topic Quiz 2 but you get 4/10 points on topic question 2 on the final exam, your final will be graded as though you got 95% = 9.5/10 on topic question 2 on the final.

Deadlines for problem sets
I will assign a small amount of extra credit for turning in each assignment by the assigned deadline. Then I will have a "late deadline," (generally 24h late) by which you can turn in a problem set with almost no loss of points. Solutions will be posted after the late deadline. One reason for this policy is that people have a variety of circumstances that cause them to miss a deadline. Rather than judging which of these circumstances merit exceptions, it seems better to me simply to have a blanket flexible policy. The reason for the (very small) difference in credit for the assigned deadline is that I want to encourage you not to fall behind on the course schedule.
If you get sick…

If you get sick, don't come to class! Your health takes priority. If you think you will miss a lot of class days, or an exam or quiz, or an assignment, let me know as soon as possible via email. The lecture notes and videos will be posted online, and I am available for Zoom appointments if you need extra reinforcement.

Academic honesty

Collaborating honestly and getting help

Collaborating and giving and getting help are encouraged. Knowing when and how to ask for help is a skill that will serve you professionally no matter what you end up doing. It has been an important skill for me in research and software engineering.

Ways you can seek help include those listed above, as well as:

- Attending office hours, even if you don't have a specific question.
- Posting discussion questions (if applicable).
- Forming study groups.

Academic dishonesty and how to avoid it

Per the Cal Poly Academic Integrity page (https://osrr.calpoly.edu/academic-integrity):

Cheating is defined as obtaining or attempting to obtain, or aiding another to obtain credit for work, or any improvement in evaluation of performance, by any dishonest or deceptive means.

It is fine to collaborate in a small group on a problem set. If you collaborate on a problem set, write up the solutions yourself. It is fine to discuss the main ideas of a proof, and even to sketch ideas or formulas on a whiteboard. However, writing up an entire proof in detail together is not fine.

These distinctions can seem blurry, and the truth is they are. A good rule of thumb is the "Kenny Loggins rule," which I learned about from TAing a class for a mentor of mine at UC Irvine: in order to make sure you are doing your own work and fully learning the material, it is a good idea, after working through a problem with a classmate, to wait at least 30 minutes, go do something completely unrelated to the class, and then write up your solution. Some people believe you also should not keep a written record of your discussions at all; personally I think this is OK, as long as you have not written up a full answer together.
ChatGPT and other generative AI

Do not use ChatGPT or similar tools for graded work in this class. I consider this academic dishonesty. I cannot physically stop you from doing so, but remember that 75% of your grade is from in-class quizzes and the final exam. If you use ChatGPT, even if you do not get caught, you will miss out on the educational benefit of working through homework problems on your own, and it will show when you get to the quizzes and the final exam.

Communication, accessibility, and inclusion

Office Hours

Instructor office hours location: Building 14 Room 223 (my office)
Office hours times: Mondays 3-4:30pm, Wednesdays 11am-12:30pm

I encourage you and welcome you to come to office hours. This is a time and space for you to get help, and only you know what help you need. Common ways in which office hours can be beneficial are:

- Getting career advice. Remember, I have software engineering experience in the nonprofit, government, and tech sectors, as well as advice to offer on academic career planning.
- Getting your questions about course material answered.
- Getting homework help.
- A place to sit and work, even if you don't have specific questions. This is your time!
- Conversations that deepen your understanding of course ideas.
- Finding out more about research you might be interested in doing.
- Letting me know what's working for you in the course and where you need more support.

Office hours can also spark collaboration between students. I have seen this happen in my office hours: students had a similar question about a homework problem, and pretty soon they were working things out on the whiteboard. This is a great way to learn! (Again: remember the Kenny Loggins rule.)

Diversity, equity, inclusion, and belonging

I would like to encourage everyone to remember that Cal Poly has a diverse student population, and this gives you the opportunity to meet peers with different identities and personal experiences, as well as a variety of prior knowledge and experiences with programming, data structures, and algorithms. For these reasons, I may randomize groups for some activities. That way you will likely get to know someone whose prior knowledge may be different from your own.

If for any reason you are uncomfortable with group activities, please email me or come and see me during office hours, and we will work together on a solution.
A note on pronouns: I will respect your pronouns. It is entirely up to you whether you want to say what your pronouns are. If you do, I will use the pronouns you specify. Mine are he/him.

I also believe in equitable class participation. I welcome enthusiastic engagement, and hope that everybody becomes comfortable speaking up. If your style is that you tend not to speak up as much, that is fine. However, I hope to create an environment where everyone feels comfortable asking questions and speaking. I also encourage you to seek help outside of class if you are feeling confused and not comfortable speaking up in class.

How to contact me

In addition to office hours and Canvas, you are welcome to email me at dfrishbe@calpoly.edu. It is my pleasure to respond to students' questions via email, schedule appointments, etc. If your question is about course material, though, I would prefer that you create a private post on Canvas instead.

Please start the subject line with [CSC 248], or I may not read the email. I am not deliberately ignoring your message, but I receive many emails, and the prefix ensures that it gets to the right folder in my inbox, which I check frequently.

Use of electronics in class

People use electronic devices in a variety of constructive ways during course time. This can include looking up relevant topics from the lecture, taking notes, writing down questions you might want to ask later, referring to the course textbook and other materials, and a number of other ways. For this reason, I am happy to have students use devices in class. We may even use these devices for in-class activities.

At the same time, I know we all have a variety of distractions competing for our attention. I trust you and encourage you to limit these distractions and focus on the constructive uses, both for your benefit and because everyone benefits from a time and space where their peers are engaging with the ideas and with each other.

Campus resources

Cal Poly provides a number of helpful campus resources (https://orientation.calpoly.edu/resources). A small sample of them are listed below that may be of interest to many students:
Tutoring

Free tutoring is available for all students taking CSC courses. We encourage you to take advantage of this resource.
https://csc.calpoly.edu/tutoring/

Career Services

The office of Career Services supports students’ career goals. Services include reviewing resumes, helping you with your job and internship search, career exploration, and guidance on changes of major.
https://careerservices.calpoly.edu/explore-services/career-counseling

Disability Resource Center

My aim is to make this course as broadly accessible as possible. I welcome you to discuss what you need to be empowered to succeed in the class, by contacting me electronically or coming to office hours.

It may also be worth contacting the Disability Resource Center. They support students with a variety of needs and forms of instructional support— which may be of help to you even if you do not explicitly require accommodations. If you do require accommodations—such as extended time for tests, etc—then the office can help you navigate these accommodations as well.
https://drc.calpoly.edu/

Transfer Center

Building 52, Room E30
https://transfercenter.calpoly.edu/ transfer@calpoly.edu

The transfer center provides academic coaching and other resources for transfer students.

Student Diversity and Belonging

Building 65, Room 217

Student Diversity & Belonging is a collaborative coalition of campus community centers serving an active role in creating culturally-rich environments at Cal Poly and in San Luis Obispo.
https://culture.calpoly.edu/ 805-756-7177 culture@calpoly.edu
Gender Equity

Student Diversity & Belonging provides support for womxn-identifying and nonbinary students, faculty, and staff, including community events in a welcoming space:
https://gec.calpoly.edu/
https://gec.calpoly.edu/AffinityGroups

CS and STEM-specific Campus Organizations

Finally, Cal Poly has a number of student organizations that may be of interest to students in this class. A few such organizations:

- Society of Women Engineers (SWE): https://www.calpolyswe.com/
- Women Involved in Software & Hardware (WISH): https://wishcalpoly.com/
- Association for Computing Machinery (ACM) Cal Poly Chapter: https://acmcalpoly.wixsite.com/acmcalpoly/about
- oSTEM: Creating a welcoming community for queer engineers and scientists https://ceng.calpoly.edu/connection/2022/08/ostem/

Final notes

The course schedule shown in this syllabus, as well as the number and topic of quizzes and problem sets, is subject to change.

Some of the materials in this course—including lecture slides—are taken from Professors Migler and Siu’s previous offerings of CSC 348.