CS 2100: Discrete Structures

Administrative Details and Syllabus
Fall 2019

Important Information

Class Website	Canvas (available through CIS)
Lectures	Tuesdays and Thursdays 12:25-1:45p in 220 ASB
Discussions	Fridays 9:40-10:30a (L120 WEB), 10:45-11:35a (2250 WEB), 11:50a-12:40p (L122 WEB), or 12:55-1:45p (L122 WEB)
Instructor	D. Erin Parker, 3144 MEB
Textbook (required)	*Discrete Mathematics: Mathematical Reasoning and Proof with Puzzles, Patterns, and Games* by Ensley and Crawley (2006, ISBN: 0471476021)
Important Dates	**Mark your calendar – quizzes* and exam may not be missed!**
Quiz 1	Tuesday, September 3 (in class)
Quiz 2	Thursday, September 19 (in class)
Quiz 3	Thursday, October 3 (in class)
Quiz 4	Tuesday, October 29 (in class)
Quiz 5	Tuesday, November 19 (in class) *Lowest quiz score will be dropped
Final exam	Thursday, December 12 (10:30a-12:30p)
Final course grade	In-class quizzes 60%, Final exam 20%, Pre-lecture Canvas quizzes 10%, Homework assignments 10%
Prerequisites	CS 1410 and MATH 1210

Course Information

CS 2100 provides an introduction to the discrete mathematics and structures that are at the foundation of computer science, as well as teaches logical thinking about discrete objects and abstract things.

Fair warning. The pacing in this class is brisk. Students should be aware that not all of the topics they need to know will be covered during lectures. Students should spend a considerable amount of time reading, watching videos, studying, and solving problems outside of lecture.

The prerequisites for this course are grades of C- or better in CS 1410: Introduction to Object-Oriented Programming and MATH 1210: Calculus I (or higher math). *Students who do not meet these prerequisites will be removed from CS 2100 in the first week of class.*

Course Materials

Website. The class website is a Canvas course available through CIS. *It is always under development*, with updates to the class schedule, course notes, homework specifications, and more, occurring regularly. It is critical that students become familiar with the class website right away and *plan to visit it several times a week, at a minimum.*

Textbook. *Discrete Mathematics: Mathematical Reasoning and Proof with Puzzles, Patterns, and Games* by Douglas E. Ensley and J. Winston Crawley (2006, ISBN: 0471476021). Students are highly encouraged to purchase used copies or rent the textbook.

Videos. Most CS 2100 topics are covered in short videos posted well ahead of each lecture. Students should watch such videos *before* the associated lecture. Regular Canvas quizzes are assigned to ensure that students prepare for each lecture by watching videos and/or reading the textbook.

Course notes. The instructor often makes use of slides, sample problems, and other materials during lecture. These items are posted on the class website following the lecture; however, such posted items may not represent completely the material covered in class. Students who must miss class are strongly encouraged to check with a classmate.

Personal computers. Students may use their own computers for completing homework assignments (typing problem solutions and taking Canvas quizzes); however, broken tools or computers, or network connectivity issues are not sufficient basis for a deadline extension. Plan ahead and use the lab computers if your own computer is not working.

Student Evaluation

In-class quizzes. Five quizzes will be given in class on 9/3, 9/19, 10/3, 10/29, and 11/19. *Make-up quizzes will not be arranged for any reason other than a documented medical emergency.* The four highest quiz scores for each student will be used to compute their final course grade; therefore, students who cannot be in attendance for one of the quiz dates above should plan to use their "drop" score accordingly. Students who cannot be in attendance for more than one of the quiz dates above should plan to take CS 2100 in a future semester.

NOTE: In-class quizzes are not the same as the Pre-lecture Canvas quizzes (see below).

Final exam. The exam is cumulative and will take place Thursday, December 12 10:30a-12:30p. *This date and time is set by the University, is not negotiable, and may not be missed.*

Pre-lecture Canvas quizzes. To ensure that students prepare adequately *before* each lecture by watching videos and/or reading the textbook, Canvas quizzes are assigned regularly.

Homework assignments. The specifications, deadline, and submission instructions for each assignment are posted on the class website. Give yourself time to think about the material. Plan on working on the assignments a little each day, and ask questions when you get stuck. Do not plan on solving an assignment all at once; it actually takes much longer to finish!

Suggested steps for approaching CS 2100 homework assignments:

1. Read the relevant sections of the textbook in a timely way.
2. Try solving the practice problems, as well as the "blue" problems that are solved for you in the back of the textbook.

3. Try solving the assigned homework problems.
4. If you are struggling with either step 2 or 3, try doing the online activities on the textbook's website (`goo.gl/JLJfLB`).
5. If you are still struggling after step 4, make use of the instructor's office hours and/or the TA help hours (see the class website for schedules).

Homework assignments are to be done independently. It is acceptable for students to discuss how to solve problems with classmates, but copying solutions is considered academic misconduct.
It is the student's responsibility to ensure the successful and timely submission of each assignment *via Gradescope* — start early and follow the instructions carefully. Corrupted or missing files are not grounds for extensions — double-check your submissions and save a digital copy of all of your work in your CADE account.

Final course grade. The final course grade is based on in-class quizzes (60%), the final exam (20%), pre-lecture Canvas quizzes (10%), and homework assignments (10%).

Regrades. Students who wish to appeal a score on a homework assignment or an in-class quiz must do so within one week of receiving the score via Gradescope.

Letter grades. The following table is used to associate numerical scores with the corresponding letter grade. Note the lack of rounding.

$93 \leq X \leq 100$	A	$87 \leq X < 90$	B+	$77 \leq X < 80$	C+	$67 \leq X < 70$	D+	$X < 60$	E
$90 \leq X < 93$	A-	$83 \leq X < 87$	B	$73 \leq X < 77$	C	$63 \leq X < 67$	D		
		$80 \leq X < 83$	B-	$70 \leq X < 73$	C-	$60 \leq X < 63$	D-		

Getting Help

To get help understanding course material, students may see the Teaching Assistant(s) during TA Help Hours, see the instructor during Office Hours, post a question to the Q&A forums on Piazza (`https://piazza.com`), or contact the course staff directly (also via Piazza). See **Important CS 2100 Information → How to get help in CS 2100** on the class website for details

Policies and Guidelines

CS 2100 laptop policy. Students are expected to engage with the instructor and classmates during class meetings. Students are permitted to use a laptop or mobile device to take notes. *Use of a laptop or mobile device for any other purpose is not permitted, and students who do so will be asked to leave the classroom.*

Other policies and guidelines. Students are bound by the following policies and guidelines:

- CS 2100 Academic Misconduct Policy `www.cs.utah.edu/~parker/2100_policy.pdf`
- School of Computing Policies and Guidelines
  `handbook.cs.utah.edu/2019-2020/Academics/policies.php`
- College of Engineering Guidelines
  `www.coe.utah.edu/students/academic-affairs/academics/semester-guidelines`

- UofU Student code
  `www.regulations.utah.edu/academics/guides/students/studentRights.html`

Students should read and understand each of these documents, asking questions as needed.

Syllabus

The following outlines the modules planned for study and the corresponding chapters in the textbook, as well as the motivation for each topic. See the class website for a detailed schedule.

Mathematical Reasoning – Chapter 1

Logic is a natural and familiar concept, giving students the opportunity to practice mathematical thinking, while getting comfortable with formalization and abstraction. Moreover, in the study of propositions and logical equivalencies, students explore Boolean expressions from a perspective that enhances their ability to use them effectively in programming.

Mathematical Writing – Chapter 2

Students learn to read and write about mathematics formally, in a variety of contexts. Giving a mathematical argument about new material tests a student's ability to reason. Furthermore, in the study of inductive proofs, students explore recursive reasoning from a perspective that enhances their ability to use recursion programatically.

Set Theory – Chapter 3

By studying sets, students exercise the problem-solving and proof-writing skills learned in previous modules. Also, students are introduced to sets as abstract mathematical structures that represent discrete objects and to the relationships between these objects. Finally, students observe how abstraction (e.g., Boolean algebra) makes certain computations (e.g., logic circuits) easier.

Functions and Relations – Chapter 4

By studying functions as binary relations, students get a perspective on functions that is different from their exposure to functions in calculus courses. By viewing a relation as another example of an abstract mathematical object, students realize that it may be thought of as a set and that all of the material from the previous module applies.

Combinatorics and Probability – Chapters 5 and 6

Students learn combinatorial analysis in order to count elements in problem solving. This knowledge is then leveraged to determine the probability of an event happening. For problems that appear difficult to solve (i.e., count), students learn how to use one-to-one correspondence from the functions module to see that two problems, which appear to be different, actually have the same solution.

Graph Theory – Chapter 7

In earlier modules, graphs are used to visualize or simplify difficult concepts (e.g., arrow diagrams for functions and relations, game trees for best-of-X series for probability). Students learn the basics of graph theory not only to better understand these previous applications, but also to lay the foundation for using graphs in other subfields of computing, such as circuit design, networking, scheduling, and more. Finally, by reading and writing proofs about graphs, the critical concept of induction is revisited.