CS 4400: Computer Systems

Administrative Details and Syllabus
Fall 2019

Important Information

Class Website	Canvas (available through CIS)
Lectures	Mondays and Wednesdays 1:25-2:45p in L101 WEB
Labs	Thursdays 8:35-9:25a, 9:40-10:30a, 10:45-11:35a, 11:50a-12:40p, 12:55-1:45p, or 2-2:50p in 3225 MEB
Instructor	D. Erin Parker, 3144 MEB
Textbook (required)	*Computer Systems: A Programmer's Perspective* by Bryant and O'Hallaron (3rd edition, 2016)
Important Dates	**Mark your calendar – exams may not be missed!**
Midterm Exam 1	Wednesday, September 25 (in class)
Midterm Exam 2	Monday, November 4 (in class)
Final exam	Thursday, December 12 (1-3p)
Final course grade	Assignments 50%, Exams 40%, Pre-lecture Canvas quizzes 10%
Prerequisites	CS 3810 and full-major status in CS or CE

Course Information

The objective of CS 4400 course is to help students bridge the gap between high-level programming and actual computer systems: processors, the memory hierarchy, operating systems, compilers, linkers, assemblers, networks, and more.

Our basic goal is to understand how a computer works, so that as programmers we can make it work efficiently. Thus, this course is an introduction to computer systems from a programmer's point of view.

The official prerequisite for this course is CS 3810 (Computer Organization). *It is strongly recommended that students complete CS 3505 (Software Practice II) before taking this class.*

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, solving problems, and programming solutions outside of lecture.

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, assignment 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. *Computer Systems: A Programmer's Perspective* by Bryant and O'Hallaron (3rd edition, 2016). Students are highly encouraged to purchase used copies or rent the textbook.

Videos. Most CS 4400 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, source code, 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 assignments 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

Assignments. The programming assignments make heavy use of C, Unix, and x86. Students not currently fluent in any of these three topics should not panic, as this course will cover them in more detail throughout the semester. However, there is an assumption that students have some familiarity with C or C++. Students should be prepared to learn some of the C programming language on their own.

To behave properly, all assignments are configured to run on a CADE Lab 1 machine. Students who choose to develop their code on any other machine are strongly encouraged to run their programs on a CADE Lab 1 machine before turning it in. *There will be no credit for programs that do not compile and run on a CADE Lab 1 machine*, even if they run somewhere else. For more information on the CADE lab and how to remotely log into these machines, see `http://www.cade.utah.edu`.

Programming assignments are due by 11:59p on the due date via Canvas. Late programming assignments are accepted according to the following rules:

- Assignments are not accepted more than 3 days after the due date.
- Assignments submitted any time X days after the due date (midnight to 11:59p) are penalized $X * 10\%$ of the assignment grade.

It is the student's responsibility to ensure the successful and timely submission of each assignment — 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.

Exams. Two midterm exams will be given in class on Wednesday, September 25 and Monday, November 4. *Makeup exams will not be arranged for any reason other than a documented medical emergency.*

The final exam is cumulative and will take place Thursday, December 12 1-3p. *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.

Final course grade. For students with an average score on exams of 65% or lower, the final course grade will be this average. Otherwise, the final course grade is based on assignments (50%), exams (40%), and pre-lecture Canvas quizzes (10%).

Regrades. Students who wish to appeal a score on an assignment or an exam must do so within one week of receiving the score.

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 4400 Information → How to get help in CS 4400** on the class website for details

Policies and Guidelines

CS 4400 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 4400 Academic Misconduct Policy `www.cs.utah.edu/~parker/4400_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 are the key topics planned for study and the corresponding chapters in the textbook. See the class website for a detailed schedule.

Getting Started – Chapter 1

Overview of computer systems

Representing Information – Chapters 2-3

Bits and bytes (information storage)
Integers (representation, arithmetic)
Floating point (representation)
x86 machine-level code (accessing information, operations)
Control flow (jumps, branches)
Procedures (run-time stack, recursion)
Data (arrays, pointers, structures, alignment)

Optimizing Code – Chapter 5

Optimizing compilers
Loops
Branch prediction
Memory performance

The Memory Hierarchy – Chapter 6

Different kinds of memory
The principle of locality
Cache memory

Running Programs on a System – Chapters 7-10

Linking
Exceptions
Processes
Signals
Virtual memory and address translation
Dynamic memory allocation
System-level I/O

Interaction Among Programs – Chapters 11-12

Network programming
Concurrent programming
Synchronization