CSCI 330: Introduction to Computer Systems

Course Information and Syllabus
Semester I, 2016–2017

Lectures	G hour: 2:00–2:50 on Mondays, Wednesdays, and Fridays
Room	Salomon 101
Lecture Notes	http://www.cs.brown.edu/courses/cs033/lectures.html A recording of each lecture will be available soon after it is given.
Text	Computer Systems: A Programmer's Perspective, 3rd Edition, Bryant and O'Hallaron, Prentice Hall 2015 (the 2nd Edition is also ok)
Prerequisite	CSCI 150, 180, or 190. In particular, you should be a competent Java programmer (though we won't be using Java)
Instructor	Tom Doeppner (twd@cs.brown.edu)
Office	CIT 405, x3-7633
Professor's Office Hours	Mondays and Wednesdays 3-4, Fridays 4-5, by appointment, or just stop by.
Head TAs	Joe Stein, Sorin Vatasoiu, Guo Wang
UTAs	Aryan Chhabria, Arun Drelich, Alejandro Molina Acosta, Abinaya Raman, Adarsh Sridhar Narayanan, Andrei Radu Voicu Moldoveanu, Benjamin Wesner, Chelse-Amoy Steele, Chirayu Poudel, Daniel Glauber, Egor Shakhnovskiy, Filip Bystricky, Giovanni Pittalis, Hans Halverson, Hangkun Ung, Joseph Bellavia, Jonathan Lister, Jared Cohen, Katherine Weingart, Lael Costa, Max Luzuriaga, Michael Xu, Nathan Hyde, Oussama Ben Abdelbaki, Peter Clarke, Peter Scott, Sarita Ballakur, Sean Segal, T Alexander Chen, Ting Xia, William Thompson, William Flotte
TA Office Hours	See http://cs.brown.edu/courses/csci0330/hours.html
Time Requirements	In addition to three hours per week in class, you will spend two hours per week in labs and 10-18 hours per week on projects. The project times will vary — less time will be required for earlier assignments than for later ones.
Goals	The primary goal is for you to understand how a modern computer system works, to the extent that you can utilize this knowledge to construct better programs. We teach you the C language and use it in most assignments because it exposes many aspects of a computer system, such as storage allocation, that are hidden when you use higher-level languages. This helps you understand what these other languages are doing for you. We teach you assembler language and high-level computer architecture so that you can appreciate what the computer is actually doing when it runs your program, and what you might do to write more

	efficient programs. We teach you the use of debugging tools, in particular gdb, not only to help you debug your code, but also to help you understand what your programs are doing. The course projects are designed to help you understand concepts such as memory allocation and concurrency in sufficient detail that you can make intelligent decisions involving these concepts in projects you pursue after the course. The lab sessions (2 hours per week) are designed to give you hands-on practice with concepts covered in class, so that you're ready to use them in the projects. The projects are time-consuming. They're intended to pull together the many concepts covered in the class and force you to think through them. Your programs will have rather subtle bugs for which you'll have to use gdb to see what they're doing. You won't simply be applying what you learn in class; you will understand the intricacies of how everything works.
All are Welcome	Our intent is that this course provide a welcoming environment for all students who satisfy the prerequisites. Our TAs have undergone training in diversity and inclusion; all members of the CS community, including faculty and staff, are expected to treat one another in a professional manner. If you feel you have not been treated in a professional manner by any of the course staff, please contact either Prof. Doeppner (the instructor), Prof. Cetintemel (the department chair), or Laura Dobler (the department's coordinator for diversity and inclusion initiatives). We take all complaints about unprofessional behavior seriously.
Clickers	The course will make use of "clickers": at each class meeting there will be one or more questions to which you must respond using your clicker.
Grading	Class participation via clickers is worth 9% of the course grade. You will get an A for answering a question correctly, a B for answering incorrectly, and no credit for not answering. Projects are given letter grades, and thus any "curving" is done on a per-project basis. Labs are given A's if done on time, C's if no more than a week late, and NC's if beyond a week late. The final course grade is the weighted average of the clicker, lab, and project grades. Each one-week project, as well as the first, is 7.27% of your final grade; each of the two other projects is 14.55% of your grade. Each lab is 1% of your final grade. Grade averages are computed using a 4-point scale: an A+ is worth 4.3 points, an A 4 points, an A- 3.7 points, a B+ 3.3 points, etc. For determining your final grade, a weighted course average of 3.5 and higher is an A, 2.5 and higher is a B, and 1.5 and higher is a C. In addition, you must pass all projects to get an A for the course; you must pass all but one one-week project to get a B for the course; you must pass all but two one-week projects or one two-week project to get a C (or S) for the course. Please note that your assignments will be graded by the TAs, all of whom are undergraduates. If you have a question about the grading of an assignment,

	please bring it up first with the TA who graded it. If your question is not resolved to your satisfaction, then bring it up with Prof. Doeppner.
Incomplete Policy	We expect everyone to complete the course on time. However, we certainly understand that there may be factors beyond your control, such as health problems and family crises, that prevent you from finishing the course on time. If you feel you cannot complete the course on time, please discuss with Prof. Doeppner the possibility of being given a grade of Incomplete for the course and setting a schedule for completing the course in the upcoming year.
Due Dates	Assignments must be handed in by 11:59 pm on their due dates. Labs are due during the last lab hours before the next lab is released.
Late Policy	The late-day policy described here applies to all late days other than those due to illness and religious holidays. Thus days missed because of job interviews are included in the late-day policy. Everyone is allowed a total of five late days on projects free of charge, but no more than three late days may be applied to any one assignment. Beyond that, you are penalized one grade level (e.g., B work goes down to a C) for each day it is late. The last project (database) must be turned in by 11:59pm, Dec. 16, regardless of how many late days you have. Your clicker scores will be based on 31 out of the 36 lectures in which clickers are used (clickers are not employed in the first lecture). Thus we will drop the five lectures on which you did most poorly in terms of your clicker responses (for example, because you were not present). No late days are allowed for labs, other than what is mentioned in the grading policy. We will apply late days to assignments in an optimal fashion (with respect to your grade). Note that late penalties are applied after grades have been curved. If you are ill, you may get an extension without using late days. Please get a note from either health services or the office of student life and contact Prof. Doeppner. If you must miss class or a project deadline because of a religious holiday, you may also get an extension without using late days, please contact Prof. Doeppner.
More Information	For more in-depth information about the course, refer to the Course Missive and Collaboration Policy linked from the course website.
Accommodations	If you feel you have physical, psychological, or learning disabilities that could affect your performance in the course, we urge you to contact SEAS (https://www.brown.edu/campus-life/support/accessibility-services/). We will do whatever we can to support accommodations recommended by SEAS.
Mental Health	Being a student can be very stressful. If you feel you are under too much pressure or there are psychological issues that are keeping you from performing well at Brown, we encourage you to contact Brown's Counseling and Psychological Services (CAPS: https://www.brown.edu/campus-life/support/counseling-and-

	psychological-services/). They provide confidential counseling and can provide notes supporting extensions on assignments for health reasons.

Lectures and Due Dates

Date	Topic	Readings	Out	Due
Sept 7	1. Intro to CSCI 330; Intro to C			
Sept 9	2. Intro to C		Lab01 – Life; Maze	
Sept 12	3. Intro to C			
Sept 14	4. Intro to C			
Sept 16	5. Intro to C			
Sept 19	6. Intro to C		Lab02 – Tools	
Sept 20			Data	
Sept 21	7. Data Representation	Chapter 2		Maze
Sept 23	8. Data Representation	Chapter 2		
Sept 26	9. x86 Assembler Language	Sections 3.1, 3.2	Lab03 – x86 Part 1	
Sept 27			Bomb	Data
Sept 28	10. x86 Assembler Language	Sections 3.4, 3.5		
Sept 30	11. x86 Assembler Language	Section 3.6		
Oct 3	12. x86 Assembler Language	Section 3.7	Lab04 – x86 Part 2	
Oct 5	13. x86 Assembler Language	Section 3.10		
Oct 7	14. Processor Arch. and Performance	Sections 5.1-5.6		
Oct 10	Holiday!			
Oct 11			Buffer	Bomb
Oct 12	15. Processor Arch. and Performance	Sections 5.7-		

		5.12		
Oct 14	16. Memory Hierarchy	Section 6.1		
Oct 17	17. Memory Hierarchy	Section 6.2	Lab05 – Profiling	
Oct 18			Performance + Strings	Buffer
Oct 19	18. Memory Hierarchy	Sections 6.3-6.5		
Oct 21	19. Architecture and OS	Sections 8.1-8.4		
Oct 24	20. Shells and Files	Section 10.1	Lab06 – Makefiles	
Oct 25			Shell Part 1	Performance + Strings
Oct 26	21. Files	Sections 10.2-10.11		
Oct 28	22. Files			
Oct 31	23. Signals	Section 8.5	Lab07 – Signals	
Nov 1			Shell Part 2	Shell Part 1
Nov 2	24. Signals	Section 8.6		
Nov 4	25. Linking and Loading	Sections 7.1-7.7		
Nov 7	26. Memory Management	Section 9.9	Lab08 – Alloc	
Nov 8			Malloc	Shell Part 2
Nov 9	27. Memory Management			
Nov 11	28. Virtual Memory	Sections 9.1, 9.2, 9.6, 9.8		
Nov 14	29. Libraries	Sections 7.12, 8.13	Lab09 – VM	
Nov 16	30. Network Programming	Sections 11.1-11.4		
Nov 18	31. Network Programming			

Nov 21	Catch-up		Lab10 – Network	
Nov 22				Malloc
Nov 23	Holiday!			
Nov 25	Holiday!			
Nov 28	32. Concurrent Programming	Chapter 12	Lab 11 – Concurrency	
Nov 29			Database	
Nov 30	33. Concurrent Programming			
Dec 2	34. Concurrent Programming			
Dec 5	35. Concurrent Programming			
Dec 7	36. Concurrent Programming			
Dec 16				Database