EE306 - Introduction to Computing

I am pleased to welcome you to your first course in Computer Engineering. I charge you to, ask questions, be curious, have fun learning and, conduct yourself with honor. I will strive to give you my best!

Quick Facts. . .

Classroom: Online: 8/25 to 9/15
UTC 3.112: from 9/20

Class Time: MW 12-1:30pm

Pre-requisites:
Credit with a grade of at least C or registration for Mathematics 408C or 408K. No formal programming experience is expected

Office Hours:
MW 2:00-3:30pm (Online)
TA Office Hours posted on Canvas

Grading Criteria:

Assessment	Percentage
8 Homeworks	20%
10 Quizzes (12 drop 2)	30%
5 Panopto Quizzes	5%
Final (Comprehensive)	15%
Programs(5)	30% (6% each)

Recitation Sessions:

Session	Time	~~Room~~ Online
17505	Fri 9-10am	~~ECJ 1.304~~
17510	Fri 10-11	~~ETC 2.102~~
17515	Fri 11-12	~~ECJ 1.304~~
17520	Fri 12-1pm	~~PMA 6.112~~

Teaching Assistants:
Jeffrey M.: jeffreymarshall@utexas.edu
Rathna S.: rathna.siva02@utexas.edu
Dasol A.: dasolahn0614@gmail.com
Euni W.: euniwu@utexas.edu

Important Dates:

Drop Deadline	Thursday 10/28
Final Exam	**Thursday 12/9 2-5pm (online)**

Class Website:
`http://canvas.utexas.edu/`

Course Overview

This is the first course in computing for students of computer engineering and electrical engineering. The objective is to provide a strong foundation that a serious student can build on in later courses across the spectrum of computer science and engineering. The idea is that a more complete understanding of the fundamentals will help a student acquire a deeper understanding of more advanced topics, whether that topic is in computer architecture, operating systems, databases, networks, algorithm design, software engineering, or whatever. The approach is "motivated" bottom-up. Starting with the transistor as a switch, we build logic gates, then more complex logic structures, then gated latches, culminating in an implementation of memory. From there, we study the computer's instruction cycle, and then a particular computer, the LC-3 (for Little Computer 3). The LC-3 captures the important structures of a modern computer, while keeping it simple enough to allow full understanding.

Textbook

Introduction to Computing Systems, 3rd edition,
by Yale Patt & Sanjay Patel
ISBN: 9781307480696,
McGraw Hill, 2019

Course Format

The content of the course will be presented in-person in a 75-min lecture twice a week. Each lecture session will also be recorded and made available through Canvas for offline viewing. There are pre-recorded videos typically no longer than 15 minutes and tackle at most three concepts of interest. There will be quizzes embedded (in the videos in-line) that test your understanding of covered material. You will be able to repeat them with no grade consequence. There will be a more targeted weekly quiz (12 in all) that is designed to keep you on track with the pacing of material covered in lecture and the assigned weekly reading. Your lowest scoring two quizzes will be dropped. Bi-weekly homeworks target, problem solving exercises from the book and elsewhere. You will be required to hand-write or type-write them electronically and upload them in the PDF format to Gradescope. Details of the process will be demonstrated in your first recitation session by your TAs.

Recitation Session

Recitation sessions are intended to reinforce and expound on topics covered in class. At times, they may present a topic in an alternate way that may be clearer to you. You are welcome to attend one, or more of the sessions. Attendance is not mandatory but strongly recommended. These sessions will be conducted by the TAs online over Zoom.

Programming Assignments

All programming assignments are in LC-3 Assembly Language. You will not only learn how to program in assembly but also how an assembler works and see and understand machine code. The assignments will focus on good programming style and practice, and teach debugging from the get go. An LC-3 Simulator allows the student to debug his/her own programs. Input (via the keyboard) and output (via the monitor) both use the physical device registers. System service routines, written in LC-3 Assembly Language, are used to perform I/O functions. They are invoked by user programs by the TRAP instruction and corresponding trap-vector. Subroutine calls and returns complete the LC-3 instruction set.

Tentative Schedule:

MONDAY	WEDNESDAY
Aug 23rd (No Class)	25th **1** Computers as Universal Computational Devices; Bits, Positional Number Systems (Ch 1,2) **Fri-Q0**
30th **2** Integers: Signed and Unsigned, Arithmetic - (Ch 2)	Sep 1st **3** Logic: AND, OR, NOT; Boolean Algebra; Floating-Point (Ch 2) **Fri-Q1**
6th Labor Day (No Class)	8th **4** More Floating Point, Other Data Types **Fri-Q2**
13th **5** Transistors, Gates - NOT, OR, NOR, AND, NAND; DeMorgans Law; TT, Logic circuits	15th **6** Combinational Logic Circuits - ADD+SUB, DECODER, PLAs **Fri-Q3**
20th **7** MUX, Full Adder, Storage Elements - RS, D Latch	22nd **8** Registers, Memory **Fri-Q4**
27th **9** Sequential Logic Circuits - Finite State Machines; Flip-flop vs. Latch	29th **10** Von Neumann Model of Computation: LC-3 ISA, Assembly vs. Machine Language (`ADD,NOT,AND,LD,ST,HALT`) **Fri-Q5**
Oct 4th **11** Problem Solving - Flowcharts; Changing Flow: (`BR`)	6th **12** The Data Path; Instruction Cycle; Reaching Farther:(`LDI, STI`) **Fri-Q6**
11th **13** Accessing related Memory (`LDR,STR`) **Program 1 Due** Tuesday 10/13 through Canvas at 11:55pm	13th **14** Data Structures - Arrays,Strings **Fri-Q7**
18th **15** Going from one to many - Loops and Arrays; Algorithms - Search and Sort (`LEA`)	20th **16** The Assembler; Data Structure - Linked-List **Fri-Q8**
25th **17** TRAP Routines, TRAP Vector Tables, The Stack **Program 2 Due** Tuesday 10/27 through Canvas at 11:55pm; (`RTI`)	27th **18** Subroutines - Library and User-defined (`JSR,JMP`) **Fri-Q9**
Nov 1st **19** Parameter passing	3rd **20** Memory-mapped I/O - Keyboard and Display **Fri-Q10**
8th **21** Data Structures - Queues **Program 3 Due** Tuesday 11/10 through Canvas at 11:55pm	10th **22** Buffer **Fri-Q11**
15th **23** I/O - Polling vs. Interrupt	17th **24** Interrupt Processing
22nd **25** **Program 4 Due** Tuesday 12/1 through Canvas at 11:55pm	24th Thanksgiving (No Class)
29th **26** The many uses of a Stack	Dec 1st **27** Buffer **FriQ12**
6th **28** Final Review **Program 5 Due** Tuesday 12/8 through Canvas at 11:55pm	8th **29** **Final Quiz - Friday 12/11**

Late Policy

Homeworks must be turned in on the due date (usually one week) in class at the beginning of class. There are no late exceptions for homeworks. Programming assignments are due midnight on the due date (one or two weeks). You are allowed a one-time exception to submit one (out of five) programming assignment late with a 10% deduction per day up to a maximum of 2 days.

Re-grading

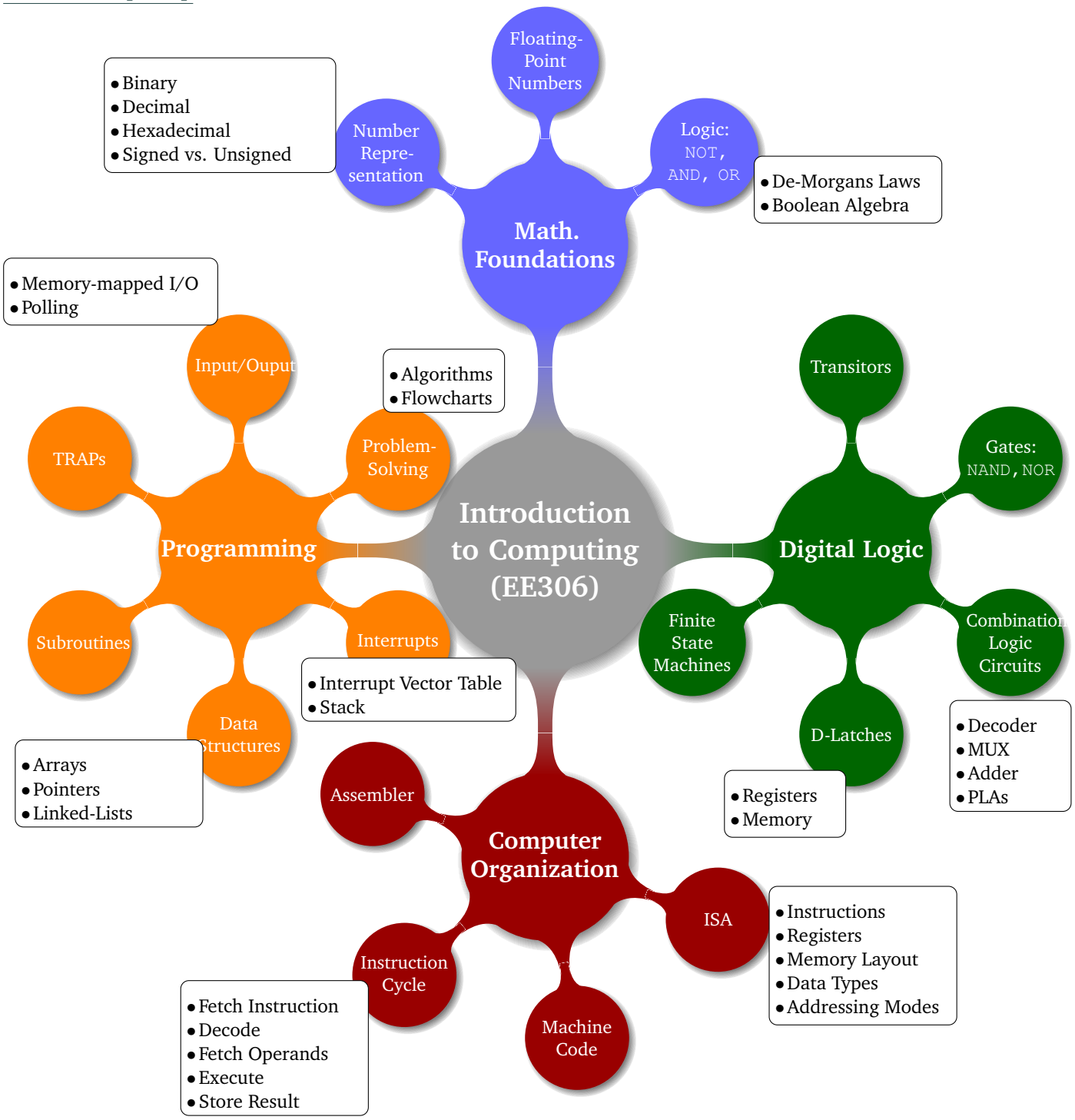
Programming assignments 1, 2 and 3 may be submitted for re-grading no later than 2 days after receiving your graded work back. The score you receive will be half the improvement you make. For example, if you make 60 on the submission and your re-submission secures you a 100, your new score will be 60+(100-60)/2 = 80.

Additional Details

The deadline for dropping without possible academic penalty is 10/28/21.

The University of Texas at Austin provides, upon request, appropriate academic adjustments for qualified students with disabilities. For more information, contact the Office of the Dean of Students at 471-6259, 471-4241 TDD, or the College of Engineering Director of Students with Disabilities, 471-4321.

EE306 Concept Map



Academic Honesty

Integrity is a crucial part of your character and is essential for a successful career. We expect you to demonstrate integrity in this course and elsewhere. In particular, your assignments must represent your own work and understanding. Academic misconduct such as plagiarism is grounds for failing the class.The following guidelines apply unless an assignment specifically states otherwise. If you have any questions about acceptable behavior, please ask the course staff. We are happy to answer your questions! You are encouraged to talk to your classmates about solution ideas, and you may reuse those ideas, but you may not examine nor reuse any other student's code. You are not allowed to copy code from any source — other students, acquaintances, the Web, etc. (Copying is forbidden via cut-and-paste, via dictation or transcription, via viewing and memorizing, etc.) You are encouraged to use books, the Internet, your friends, etc. to get solution ideas, but you may not copy/transcribe/transliterate code: get the idea, close the other resource, and then (after enough time that the idea is in your long-term, not short-term, memory) generate the code based on your own understanding.

Examining other people's code

You may sometimes find it useful to do a web search to find snippets of code that perform some particular operation, and you may subsequently paste this code into your own program. This can be an acceptable short-term strategy if it helps you get past a particular roadblock. However, you must later go back, remove the code you did not write yourself, and write the replacement on your own, from scratch. It is your responsibility to understand everything that you turn in. We reserve the right to ask you to explain any part of your work. If you are not able to explain what it means and why you chose it, that is presumed evidence of copying/cheating.

Later, when you are writing your own programs after you complete this course and your degree, it's fine to copy others' code if the license associated with the code permits such use. However, in your future career, please remember two things:

1. It is your ethical duty to properly cite the source of any code that you did not write yourself. Give credit where credit is due.
2. You should still understand any code that you copy. Otherwise, if and when the code does not work (for example, if the original author made an assumption that is not true in your program), you will lose more time debugging than you saved by copying.

The key idea is that we want you to understand. Sometimes you can achieve that by examining and understanding other people's code. But you can never achieve that by copying alone. We are committed to preserving the reputation of your UT degree. To guarantee that every degree means what it says it means, we must enforce a strict policy on academic honesty: every piece of work that you turn in with your name on it must be yours. As an honest student, you are responsible for enforcing this policy in three ways:

1. You must not turn in work that is not yours, except as expressly permitted by the instructors. Specifically, you are not allowed to copy someone else's program code. This is plagiarism.
2. You must not enable someone else to turn in work that is not his or hers. Do not share your work with anyone else. Make sure that you adequately protect your files. Even after you have finished a class, do not share your work or published answers with students who come after you. They need to do their work on their own.
3. You must not allow someone to openly violate this policy because it diminishes your effort as well as that of your honest classmates.

Students who violate University rules on scholastic dishonesty in assignments or exams are subject to disciplinary penalties, including the possibility of a lowered or 0 grade on an assignment or exam, failure in the course, and/or dismissal from the University. Changing your exam answers after they have been graded, copying answers during exams, or plagiarizing the work of others will be considered academic dishonesty and will not be tolerated. Plagiarism detection software will be used on the programs submitted in this class.

Land Acknowledgment

I/we would like to acknowledge that we are meeting on Indigenous land. Moreover, I/we would like to acknowledge and pay our respects to the Carrizo & Comecrudo, Coahuiltecan, Caddo, Tonkawa, Comanche, Lipan Apache, Alabama-Coushatta, Kickapoo, Tigua Pueblo, and all the American Indian and Indigenous Peoples and communities who have been or have become a part of these lands and territories in Texas, here on Turtle Island.

Class Recordings: Class recordings are reserved only for students in this class for educational purposes and are protected under FERPA. The recordings should not be shared outside the class in any form. Violation of this restriction by a student could lead to Student Misconduct proceedings.

If cheating is discovered, a report will be made to the Dean of Students. Allegations of Scholastic Dishonesty will be dealt with according to the procedures outlined in Appendix C, Chapter 11, of the General Information Bulletin, `http://www.utexas.edu/student/registrar/catalogs/`

Sharing of Course Materials is Prohibited

No materials used in this class, including, but not limited to, lecture hand-outs, videos, assessments (quizzes, exams, papers, projects, homework assignments), in-class materials, review sheets, and additional problem sets, may be shared online or with anyone outside of the class unless you have my explicit, written permission. Unauthorized sharing of materials promotes cheating. It is a violation of the University's Student Honor Code and an act of academic dishonesty. I am well aware of the sites used for sharing materials, and any materials found online that are associated with you, or any suspected unauthorized sharing of materials, will be reported to Student Conduct and Academic Integrity in the Office of the Dean of Students. These reports can result in sanctions, including failure in the course.

Class Recordings

Class recordings are reserved only for students in this class for educational purposes and are protected under FERPA. The recordings should not be shared outside the class in any form. Violation of this restriction by a student could lead to Student Misconduct proceedings.

COVID Caveats

To help keep everyone at UT and in our community safe, it is critical that students report COVID-19 symptoms and testing, regardless of test results, to University Health Services, and faculty and staff report to the HealthPoint Occupational Health Program (OHP) as soon as possible. Please see this link to understand what needs to be reported. In addition, to help understand what to do if a fellow student in the class (or the instructor or TA) tests positive for COVID, see this University Health Services link.