ECE406/H - 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!

Class Website:
`https://utexas.instructure.com/courses/1454931`

Classrooms+Time:
18461, 18462, 18463, 18464 - BUR 112 (MW 3:00-4:30pm)
18446, 18448, 18449, 18451 - WAG 101 (TTh 11-12:30pm)
18481, 18482, 18483, 18484 - CPE 2.204 (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:
TTh 9:30-11:00am; 2:00-4:00pm (Room:**EER 5.824**)
TA Office Hours posted on Canvas

Grading Criteria:

Assessment	Percentage
8 Homeworks	10%
10 Quizzes (drop 2)	5%
Midterm Exam 1(Paper)	20%
Midterm Exam 2(Online)	20%
Final (Comprehensive)	20%
Programs(5)	20%
Recitation Activity	5%

Recitation Sessions:

Session	Time	Room
18461	Thu 5-6pm	ETC 2.102
18462	Thu 4-5pm	ETC 2.102
18463	Fri 3-4pm	ECJ 1.222
18464	Fri 2-3pm	ECJ 1.222
18446	Thu 3-4pm	ECJ 1.304
18448	Fri 4-5pm	ECJ 1.304
18449	Fri 3-4pm	ECJ 1.304
18451	Fri 2-3pm	ECJ 1.304
18481	Fri 11am-12pm	ETC 2.102
18482	Fri 12-1pm	ECJ 1.304
18483	Fri 12-1pm	PMA 5.112
18484	Fri 1-2pm	PMA 5.112

Teaching Assistants:
Roy M.:roy.mor@utexas.edu; Yahir L.:yel232@my.utexas.edu; Shaun K.:roy.mor@utexas.edu; Amina M.: amina.meddad@utmail.utexas.edu; Arsen I.:arsen_iman@utexas.edu; Jeremy S.:jsheng555@utmail.utexas.edu; Emma W.:ew24223@my.utexas.edu; Kavya R.: r.e.kavyasri@utexas.edu; Mircea N.:mnt884@my.utexas.edu

Important Dates/Times/Location:

Midterm Exam 1 (Paper-based)	**Thu 10/1 7-9pm UTC 2.112A**
Midterm Exam 2 (Computer-based)	**Thu 11/5 7-9pm UTC 2.102A**
Final Exam (Paper-based)	**Fri, 12/11 3:30-6:30 pm TBD**

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

Longhorn Textbook Access (LTA): Is a new initiative between UT Austin, The University Co-op and textbook publishers to significantly reduce the cost of digital course materials for students. You are automatically opted into the program but can easily opt-out (and back in) via Canvas through the 12^{th} class day. If you remain opted-in at the end of the 12^{th} class day you will receive a bill through your "What I Owe" page and have until the end of the 18^{th} class day to pay and retain access. If you do not pay by the 18^{th} class day, you will lose access to the materials after the 20^{th} class day and your charge will be removed. More information about the LTA program is available at:
https://www.universitycoop.com/longhorn-textbook-access

Course Format

The content of the course will be presented in-person in a 75-min lecture twice a week. There will be a more targeted weekly quiz (12 in all) 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 a PDF format to Gradescope. Details of the process will be demonstrated in your first recitation session by your TAs. All three exams are common (to both sections) exams conducted outside of class time (see ←). The first midterm is paper-based over content covered in the first third of the semester. The second midterm is a computer-based programming exam. The end-of-semester final exam is a paper-based comprehensive exam with greater emphasis on content covered post first midterm.

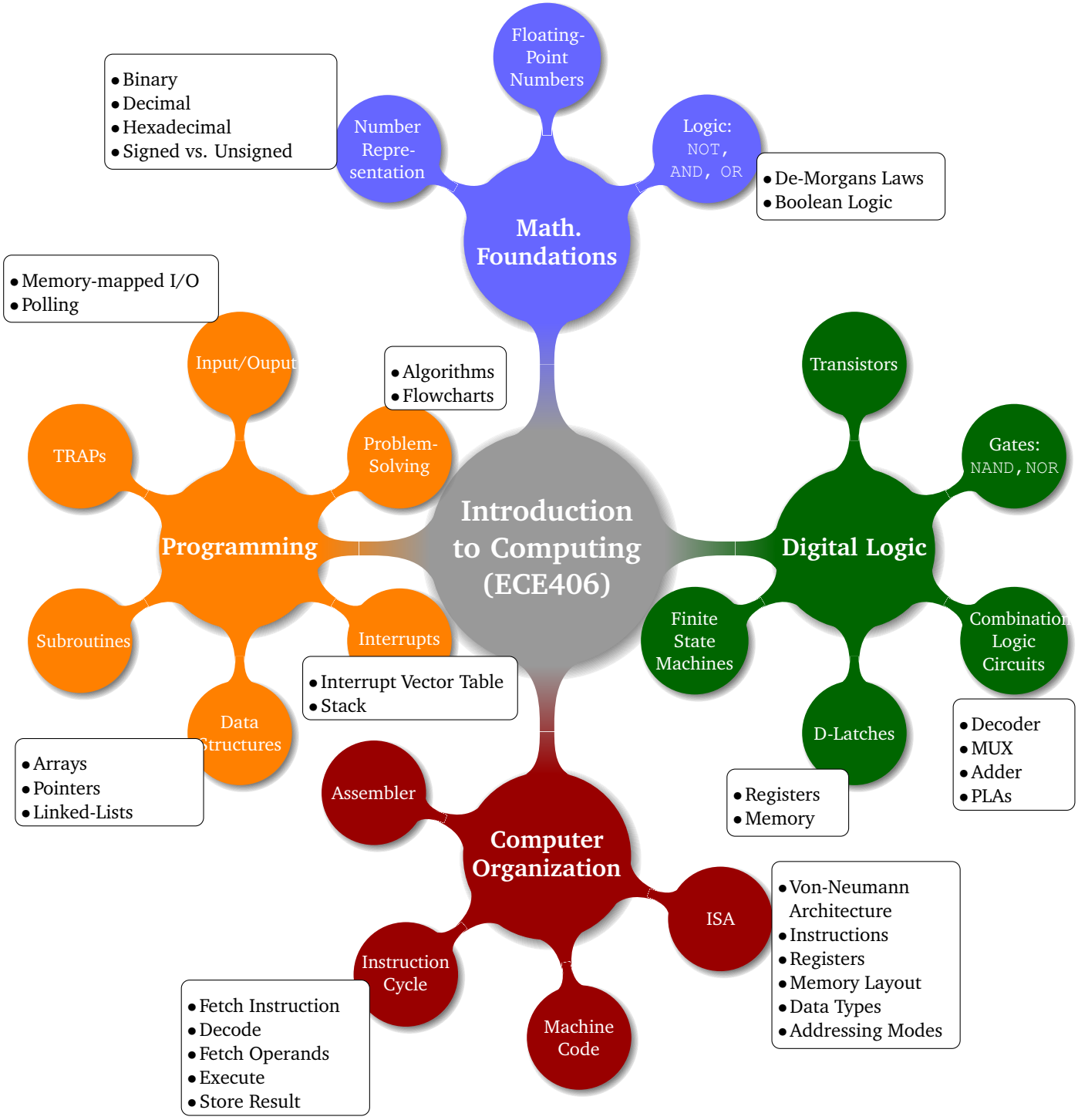
Recitation Session

Recitation sessions are TA run and 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 mandatory.

Tentative Schedule (Dr. Valvano's class meets TTh):

MONDAY	WEDNESDAY
Aug 24th **1** Computers as Universal Computational Devices; Bits, Positional Number Systems (Ch 1,2)	26th **2** Integers: Signed and Unsigned, Arithmetic - (Ch 2) **Fri-Q0**
31st **3** Logic: AND, OR, NOT; Boolean Algebra; Floating-Point (Ch 2)	Sep 2nd **4** More Floating Point, Other Data Types **Fri-Q1**
7th Labor Day (No Class)	9th **5** Transistors, Gates - NOT, OR, NOR, AND, NAND; DeMorgans Law; TT, Logic circuits **Fri-Q2**
14th **6** Combinational Logic Circuits - ADD+SUB, DECODER, PLAs	16th **7** MUX, Full Adder, Storage Elements - RS, D Latch **Fri-Q3**
21st **8** Registers, Memory	23rd **9** Sequential Logic Circuits - Finite State Machines; Flip-flop vs. Latch **Fri-Q4**
28th **10** Von Neumann Model of Computation: LC-3 ISA, Assembly vs. Machine Language (`ADD,NOT,AND,LD,ST,HALT`)	30th **11** Problem Solving - Flowcharts; Changing Flow: (`BR`) **Thu 10/1 Midterm 1**
Oct 5th **12** The Data Path; Instruction Cycle; Reaching Farther:(`LDI, STI`)	7th **13** Accessing related Memory (`LDR,STR`) **Fri-Q5**
12th **14** Data Structures - Arrays,Strings **Program 0 Due** Monday 10/12 on Gradescope	14th **15** Loops and Arrays; Algorithms - Search and Sort (`LEA`) **Fri-Q6**
19th **16** The Assembler; Data Structure - Linked-List **Program 1 Due** Monday 10/19 on Gradescope	21st **17** TRAP Routines, TRAP Vector Tables, The Stack (`RTI`) **Fri-Q7**
26th **18** Subroutines - Library and User-defined (`JSR,JMP`)	28th **19** Parameter passing **Fri-Q8**
Nov 2nd **20** Memory-mapped I/O - Keyboard and Display **Program 2 Due** Monday 11/2 on Gradescope	4th **21** Data Structures - Queues **Thu 11/5 Midterm 2**
9th **22** I/O - Polling vs. Interrupt	11th **23** Interrupts **Fri-Q10**
16th **24** Interrupt Processing Data Path **Program 4-1 Due** Monday 11/16 on Gradescope **Program 4-2 Due** Sunday 11/22 on Gradescope	18th **25** The many uses of a Stack **Fri-Q11**
23rd Thanksgiving (No Class)	25th Thanksgiving (No Class)
30th **26** In class Program 5 coding	Dec 2nd **27** In class Program 5 coding **Fri-Q12**
7th **28** Final Review **Program 5 Due** Monday 12/7 on Gradescope	9th **29** **Final Exam - Fri 12/11**

EE306 Concept Map



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. 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 11/18/26.

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.

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.

Exams
The mid-term exams are at common exam time but different exams for the two sections I teach. First mid-term is on Thursday 10/1 from 7 to 9pm; The second is on Thursday 11/5 from 7 to 9pm. The syllabus for the mid-term exams will be posted on the class Canvas site. The final exam will be held Friday, 12/11 at 3:30pm. If you have a time conflict you may take the exam on the makeup date which is Saturday, 12/12 at 3:30pm The final is comprehensive and includes the entire 10 chapters though emphasis will be more on the later chapters.

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.

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.

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.

Examining other's code

You may sometimes find it useful to get snippets of other's code (from friends, online or using an AI tool) 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:

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.

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.

Emergency Evacuation Procedures

The following recommendations regarding emergency evacuation from the Office of Campus Safety and Security, 512-471-5767, `http://www.utexas.edu/safety/`. Occupants of buildings on The University of Texas at Austin campus are required to evacuate buildings when a fire alarm is activated. Alarm activation or announcement requires exiting and assembling outside.

- Familiarize yourself with all exit doors of each classroom and building you may occupy.
- Students requiring assistance in evacuation shall inform their instructor in writing during the first week of class.
- In the event of an evacuation, follow the instruction of faculty or class instructors. Do not re-enter a building unless given instructions by the following: Austin Fire Department, The University of Texas at Austin Police Department, or Fire Prevention Services office.
- Link to information regarding emergency evacuation routes and emergency procedures can be found at: `http://www.utexas.edu/emergency`