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: EER 1.516

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 2:00-4:00pm (EER 5.824)
TA Office Hours on Canvas

Grading Criteria:

Assessment	Percentage
Homeworks	20%
Mid-terms(2)	30% (15% each)
Final Exam	20%
Programs(5)	30% (6% each)

Recitation Sessions:

Session	Time	Room
16060	Tue 10-11am	SZB 380
16065	Tue 11-Noon	ETC 2.132
16070	Tue Noon-1pm	ETC 2.132
16075	Tue 5-6pm	GDC 4.302

Teaching Assistants:

Prakash L.:	prakashluu@utexas.edu
Ramesh A.:	arjun99ramesh@gmail.com
Matthew B.:	mebarondeau@utexas.edu

Important Dates:

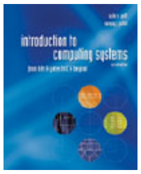
Midterm 1	Monday 10/8
Midterm 2	Monday 11/19
Drop Deadline	Thursday 11/7
Final Exam	Saturday 12/15 2-5pm (Location:TBD)

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, 2nd edition,
by Yale Patt & Sanjay Patel
ISBN-13: 978-0-07-246750-5,
McGraw Hill, 2003

Recitation Session

Recitation sessions are intended to reinforce and expound on topics covered in class. TAs conducting these sections are competent to answer any questions you have regarding the material 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.

Programming Assignments

The first programming assignment is in the machine language of the LC-3. From there, we move up to Assembly Language, and learn how an assembler works. The remaining programming assignments are in LC-3 Assembly Language. We cover 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 will be held during regular class time. The syllabus for the mid-term exams will be posted on the class Canvas site. The final exam will be held according to university schedule. The final is comprehensive and includes the entire 10 chapters though emphasis will be more on the later chapters.

Tentative Schedule:

MONDAY	WEDNESDAY
Aug 27th (No Class)	29th **1** Hardware vs. Software; Computers as Universal Computational Devices; The course journey - (Ch 1)
Sep 3rd Labor Day (No Class)	5th **2** Bits, Integer representations, Bit Operations - (Ch 2)
10th **3** Logical Operations, AND, OR, NOT; Other Data Types (Ch 2)	12th **4** Transistors, Gates - NOT, OR, NOR, AND, NAND; DeMorgans Law
17th **5** Combinatorial Logic Circuits - Full ADDER, MUX, DECODER, PLAs	19th **6** Storage Elements - RS, D Latch, Register, Basic concept of Memory
24th **7** Sequential Logic Circuits - Finite State Machines	26th **8** Von Neumann Model of Computation: Fetch-Decode-Execute-Store
Oct 1st **9** ISA - Introduction to Little Computer 3 (LC-3), ISA, Problem Solving and Debugging	3rd **10** Buffer/Exam 1 Review
8th **11** **First Mid-term Exam** (Syllabus - Chapters 1,2,3,4)	10th **12** Problem Solving and Debugging
15th **13** A Stored Program in LC-3 ISA **Program 1 Due** Tuesday 10/16 through Canvas at 11:55pm	17th **14** Assembly Language and The Assembler
22nd **15** Detailed example in LC-3 Assembly language	24th **16** Physical I/O
29th **17** TRAP Routines, TRAP Vector Tables **Program 2 Due** Tuesday 10/30 through Canvas at 11:55pm	31st **18** Subroutines - Library and User-defined
Nov 5th **19** Stacks. Parameters. How are they passed	7th **20** Example Applications
12th **21** Example Applications **Program 3 Due** Tuesday 11/13 through Canvas at 11:55pm	14th **22** Catch up, Review
19th **23** **Second Mid-term Exam** (Syllabus - Chapters 5,6,7,8 and 9)	21st Thanksgiving (No Class)
26th **24** Interrupt Processing	28th **25** Interrupt Processing
Dec 3rd **26** ASCII/binary Conversion **Program 4 Due** Tuesday 12/4 through Canvas at 11:55pm	5th **27** Stack use for Arithmetic expression parsing; The Calculator Example (Putting it all together)
10th **28** Final Exam Review **Program 5 Due** Tuesday 12/11 through Canvas at 11:55pm	12th **29**

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

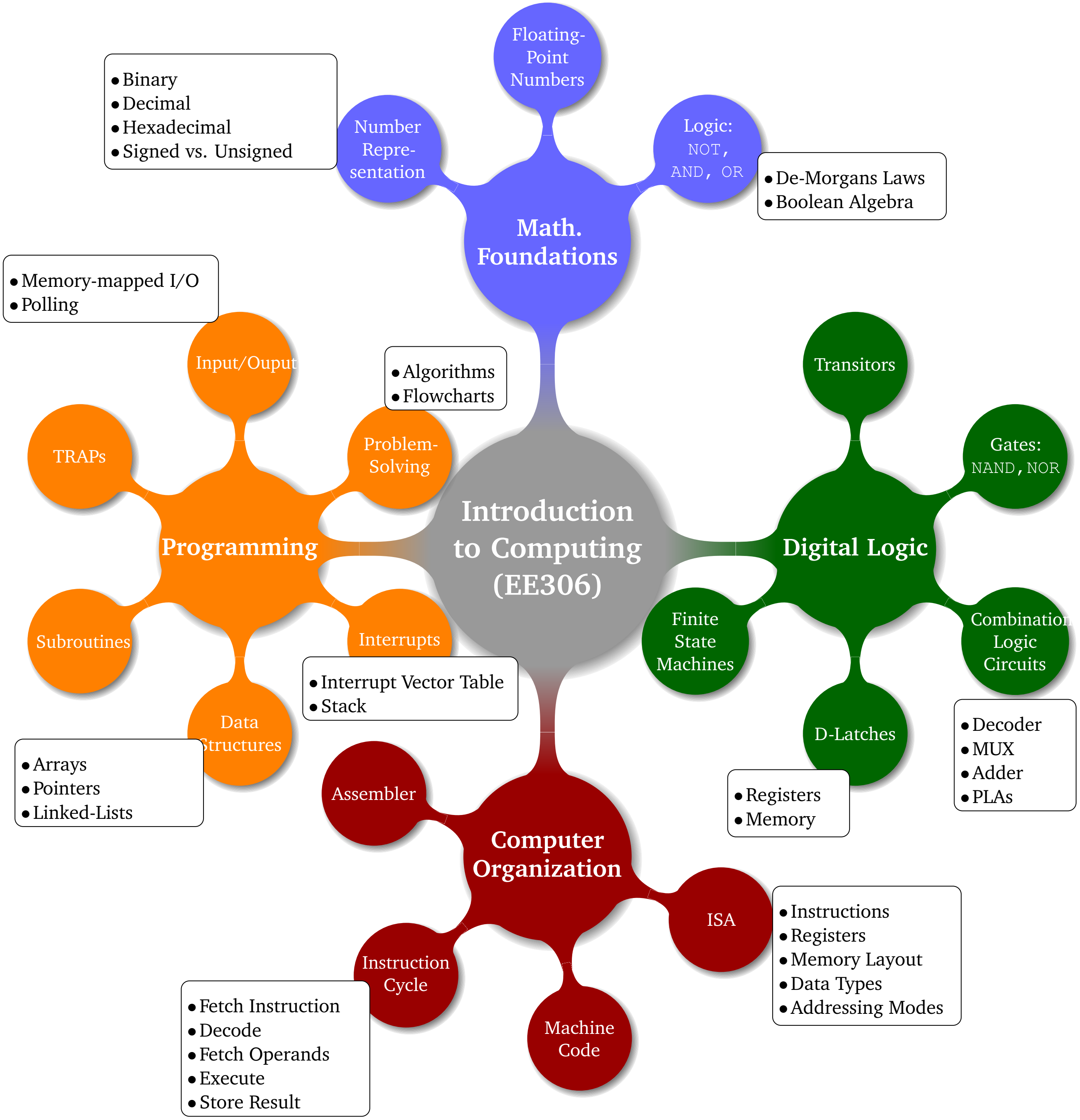
Programming assignments 1, 2 and 3 may be submitted for re-grading no later than 2 days after receiving your graded work back. The score you receive will be half the improvement you make. For example, if you make 60 on the submission and your re-submission secures you a 100, your new score will be 60+(100-60)/2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

Additional Details

The deadline for dropping without possible academic penalty is 11/7/18.

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.

Concept Map


Introduction to Computing (EE306)
Math. Foundations
Floating-Point Numbers
Number Repre-sentation
• Binary
• Decimal
• Hexadecimal
• Signed vs. Unsigned
Logic: NOT, AND, OR
• De-Morgans Laws
• Boolean Algebra
Programming
Input/Ouput
• Memory-mapped I/O
• Polling
Problem-Solving
• Algorithms
• Flowcharts
TRAPs
Subroutines
Interrupts
• Interrupt Vector Table
• Stack
Data Structures
• Arrays
• Pointers
• Linked-Lists
Digital Logic
Transitors
Gates: NAND,NOR
Finite State Machines
Combinatio Logic Circuits
• Decoder
• MUX
• Adder
• PLAs
D-Latches
• Registers
• Memory
Computer Organization
Assembler
ISA
• Instructions
• Registers
• Memory Layout
• Data Types
• Addressing Modes
Instruction Cycle
• Fetch Instruction
• Decode
• Fetch Operands
• Execute
• Store Result
Machine Code

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 homework assignment. 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.

In summary, 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.

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/`