Department of Electrical and Computer Engineering

The University of Texas at Austin

EE 306, Introduction to Computing
The FIRST Computing Course for EE and CE Majors
Unique Numbers 15900, 15905, 15910, 15915, 15920, 15925, 15930, 15935
Fall, 2019

Instructor: Yale Patt

Web: http://www.ece.utexas.edu/~patt
Office: 5.802 Engineering Education and Research Center (EERC)
Phone: 512-471-4085
Email: patt@ece.utexas.edu
Office hours: MW, 5:00pm to 6:30pm, at the front of the classroom or in my office (EER 5.802), and by appointment

Important information about EE 306 is also contained in a handout titled: Introduction to EE 306. Please be sure to read it. It expands on a number of items in this Course Descriptor. Also, make sure you read the handout titled: Course Outline (aka, Syllabus). It contains a lecture by lecture schedule of topics for the entire semester, the major emphasis of each discussion session, the due dates of problem sets and programming assignments, and the dates of all examinations.

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 early in your education will help you acquire a deeper understanding of more advanced topics later, whether that topic is in computer architecture, operating systems, data base, networks, algorithm design, software engineering, or whatever. I call the approach "motivated" bottom-up. That is, after providing some overview of why a new concept is important, we attempt to tie that new concept to what you already understand. 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 and a finite state machine. From there, we study the computer's instruction cycle, and then a particular computer, the LC-3 (for Little Computer 3). Why "3"? ...because we got it wrong the first couple of times! The LC-3 captures the important structures of a modern computer, while keeping it simple enough to allow complete understanding. 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 gitgo. An LC-3 Simulator allows the student to debug his/her own programs. Input (via the keyboard) and output (via the monitor) both use 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.
Course Description:	The course will cover the material of Chapters 1 through 10 of the textbook. See the handout, Course Outline (aka Syllabus), available in hard copy and on the class web site.
Meeting Info:	The course consists of three hours of lecture + a one hour discussion section each week. Lectures will be in WCH 1.120, MW 3:30pm - 5:00pm. Discussion sections are scheduled at various times of the day on Thursdays and Fridays. The Course website lists the meeting times and room numbers for each of the 8 discussion sections. Note that each discussion section has its own unique id. Students are free to attend the discussion section of their choice, although I would like you to attend your assigned discussion section during the first week of the course.
Teaching Assistants:	Sabee Grewal, Arjun Ramesh, Joseph Ryan, Chirag Sakhuja, Meiling Tang, Grace Zhuang
Course Home Page:	http://www.ece.utexas.edu/~patt/19f.306
Textbook:	*Introduction to Computing Systems: from bits and gates to C, C++ and beyond*; Yale N. Patt and Sanjay J. Patel; Mc-Graw Hill, 2019, 3rd edition. You will need the 3rd edition. There have been many substantial changes since the second edition.
Additional Course Resources:	Class handouts will be supplied when necessary to supplement the concepts discussed in lecture. Other information will be downloadable from the course homepage.
Prerequisites:	There are no computer pre-requisites. No programming experience is assumed. On the other hand, we do assume that the student is able and highly motivated to learn and has the energy and intelligence to support that motivation. We also assume that your mathematics background is at least at the level where you are enrolled in a strong calculus sequence. EE 306 is intended for freshmen, but is open to all students who want a serious introduction to computing in general and computer engineering in particular.

If you are one of those with no experience using computers, please do not be intimidated by those in the classroom with years of computer experience. It has been the case many times that students with no computer experience earn A in the course, and students with lots of experience earn C or lower.

Homework Policy: Problem sets will be assigned periodically. Usually, students will have between one and two weeks to complete them. Students are encouraged to form study groups to work homework problems. Only one copy of a problem set per group need be turned in. There will be five programming assignments, one in machine language, four in assembly language. Dates and times when problem sets and programming assignments are due are contained on the Course Outline. Students are encouraged to discuss the structure of the program with other members of their group. However, the detailed algorithm and actual coding must be done by the student working alone. Collaboration on the algorithm and/or giving or accepting actual code for a program constitutes cheating, and will be dealt with harshly. **Please see the handout: Introduction to EE 306 for more information on what constitutes cheating.**

Quiz and Exam Policy: There will be two exams in class, one on October 16, the other on November 20. There will be a final exam during the normal final exam period. Our preliminary information has the Final Exam scheduled for December 13 from 7 to 10pm. However, the university sometimes changes the dates of some final exams, so it is important to check the final exam schedule when it is formally posted. All exams will be closed book, with two exceptions: (1) The student may bring into the exam three sheets of paper on which the student is free to write anything he/she wishes. All three sheets must be original sheets in the student's own handwriting. (2) The student may bring into the exam any handouts that have been expressly permitted by the instructor prior to the exam. The student may not have in his possession during any exam a calculator, a mobile phone, or any other mechanism that in the view of the instructor can distract from a fair and balanced examination

Final Exam: See above.

Grading Mechanics: Course Grade:

- Problem sets (2% each, times 5 assignments = 10%)
- Programming Assignments (5% each, times 5 assignments = 25%)
- Two in-class exams (17% each = 34%)
- Final exam (25%)
- Other (6%)

Policy: Problem sets and programming assignments are due on the date and at the time specified. Make-up exams will not be given, except under very rare

circumstances. Excused absence from an in-class exam must be obtained in advance, except in very rare circumstances.

The MEC Common Evaluation form will be used to evaluate the instructor in this course.

Additional details:

The deadline for dropping without possible academic penalty is October 31, 2019.

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/.

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.

Finally, welcome to EE 306. I hope you enjoy your time in it.

Department of Electrical and Computer Engineering
The University of Texas at Austin

EE 306, Fall, 2019
Yale Patt, Instructor
TAs: Chirag Sakhuja, Sabee Grewal, Grace Zhuang, Arjun Ramesh, Joseph Ryan, Meiling Tang
Course Outline
August 28, 2019

August 28: Lecture 1. Overview of EE 306.
--The computer -- a complex system organized in levels of interpretation.
--The computer -- a universal computational device; given enough time and space it can do anything any other computational device does.

August 29,30: Discussion Session. Orientation to the LRC system, tools.

September 2: Labor Day. University closed. No class.

September 4: Lecture 2: Bits and operations on bits.
-- The bit as a unit of information.
-- Encoding of bits: Binary numbers (integer data type, ASCII characters.
-- Negative numbers, 2's complement representation, sign-extension.
-- hex representation of binary numbers.
-- Arithmetic operations on numbers. ADD, SUB. [Note that x+x = left shift]
-- Logical operations on bits. AND, OR, NOT.

September 5,6: Discussion Session. Emphasis on Chapters 1,2, problem set 1.

September 9: Lecture 3. Bits and operations on bits (continued).

September 11: Lecture 4. Basic Logic Structures
-- The transistor as a switch
-- Basic Gates (AND, OR, NOT)
-- Truth table representations
-- Any arbitrary function can be built out of these gates (no attempt at minimization. Just an awareness exercise)
-- full ADDER, MUX, DECODER
-- Basic storage element (Gated RS latch)
-- A register

September 12,13: Discussion Session. Emphasis on Chapter 2, problem set 1

Problem set 1, due before class, September 16.

September 16: Lecture 5. Basic Logic Structures (continued).

September 18: Lecture 6. Memory and Finite State Machines
-- a logic circuit to implement a small piece of memory (perhaps 2**2 x 3)
-- concept of memory: address space, addressability
-- The notion of state (one of the most important concepts in engineering)
-- State diagram, Next State table, State Assignment
-- Implementation example: sequential machine

September 19,20: Discussion Session. Emphasis on Chapter 3, problem set 2

September 23: Lecture 7. Memory and Finite State Machines (continued).

Problem set 2, due before class, September 25.

September 25: Lecture 8. Introduction to Von Neumann model and the LC-3 ISA.
-- the basic structure of the Von Neumann model, showing the basic flow.
-- instruction = opcode, operands
-- encoding of instructions and data
-- instruction cycle (Fetch, Decode, EA, Fetch data, Execute, Store result)
-- organization of memory
-- address space, addressability revisited (MAR, MDR)
-- instruction formats
-- operate, data movement, and control instructions
-- LD/ST (also, indirects)
-- control (condition codes: N,Z,P)
-- The datapath necessary to implement the LC-3
-- I/O via the TRAP instruction [Keyboard in, screen out]

September 26,27: Discussion Session: Intro to LC-3, the Simulator and Program 0.

Programming Lab 0 due, 11:59pm, September 29.

September 30: No lecture. Extra office hours in lieu of class.

October 2: Lecture 9. The LC-3 Data Path, A more sophisticated LC-3 program
-- a detailed example in machine language
-- example will use keyboard input, monitor output.
-- example will include entering data via the keyboard and outputting on the monitor

October 3,4: Discussion Session: Chapter 5, PL1

Programming Lab 1 due, 11:59pm, October 6.

October 7, Lecture 10. Structured Programming and Debugging
-- Elements of Problem Solving (stepwise refinement, systematic decomposition, etc.)
-- Fundamentals of Debugging (setting breakpoints, single-step, deposit, examine, etc.)
-- the control structure of a stored program (sequential, conditional, iteration)

October 9: No class. Extra office hours in lieu of class

October 10,11: Discussion Session: Prepare for Midterm exam.

Problem set 3, due before class, October 14.

October 14: Lecture 11. Review or catch up!

October 16: Lecture 12. **Exam 1.**

October 17,18: Discussion Session: Go over exam, discuss Programming Lab 2.

October 21: Lecture 13. Moving up a level, Assembly Language and the Assembler.

October 23: Lecture 14. Detailed examples in Assembly Language.
-- going from higher to lower level: interpretation vs. translation
-- translation: what do assemblers and compilers do?
-- hand assemble programs from earlier lectures.
-- revisiting the character count problem

October 24,25: Discussion Session: Emphasis on Chapter 7 and Programming Lab 2.

Programming Lab 2 due, 11:59pm, October 27.

October 28: Lecture 15. JSR/RET, Stack
-- saving/restoring state
-- success/failure mechanisms

October 30: Lecture 16. Queues, Linked Lists, Character Strings

October 31, November 1: Discussion Session: Problem set 4, Programming Lab 3

Problem set 4, due before class, November 4.

November 4: Lecture 17. Recursion.

November 6: Lecture 18. Recursion (continued), Trees

November 7,8: Discussion Session: Data Structures, Programming Lab 3

Programming Lab 3 due, 11:59pm, November 10.

November 11: Lecture 19. Physical I/O.
-- asynchronous activity
-- memory mapped vs. special I/O instructions
-- program control vs. device (interrupt) driven
-- device registers (KBDR, KBSR, DDR, DSR)
-- Synchronization via the ready bit.
-- interrupt enable bit
-- I/O Service Routines

November 13: Lecture 20. Physical I/O, continued.

November 14,15: Discussion Session: Prepare for Midterm 2, Programming Lab 4

Problem set 5, due before class, November 18.

November 18: Lecture 21. Review or catch up.

November 20: Lecture 22. **Exam 2.**

November 21,22: Discussion Session: Go over midterm, Programming Lab 4

Programming Lab 4 due, 11:59pm, November 24.

November 25: Lecture 23. TRAPs and Interrupts

November 27: No class, Thanksgiving Recess. Enjoy the Holiday!

December 2: Lecture 24. TRAPs, Interrupts (continued).

December 4: Lecture 25. The Calculator and what comes after EE306
-- ASCII/2's-complement conversion
-- Stack arithmetic
-- The Calculator, itself
-- Parallelism. The latest hot button!
-- Preview of coming attractions: The ARM ISA

December 5,6: Discussion Session: Programming Lab 5, Prepare for Final Exam.

Programming Lab 5 due, 11:59pm, December 8.

December 9: Lecture 26. Any OTHER questions!

Problem set 6, not to be handed in, use for final exam preparation.

December 13. **Final Exam**, 7 to 10pm.
(according to the Registrar's Course Schedule, which **he can change**.)

Programming Labs:

1st programming Lab (machine language) -- Due: October 6, 11:59pm.
2nd programming Lab (assembly language) -- Due: October 27, 11:59pm.
3rd programming Lab (assembly language) -- Due: November 10, 11:59pm.
4th programming Lab (assembly language) -- Due: November 24, 11:59pm.
5th programming Lab (assembly language) -- Due: December 8, 11:59pm.

Problem Sets:

1st problem set, (emphasis on Chapters 1,2). Due: just before class, September 16.
2nd problem set, (emphasis on Chapter 2,3). Due: just before class, September 25.
3rd problem set, (emphasis on Chapter 4,5). Due: just before class, October 14. (Note: exam on October 16)
4th problem set, (emphasis on Chapter 7,8). Due: just before class, November 4
5th problem set, (emphasis on Chapters 8,9). Due: just before class, November 18. (Note: exam on November 20)
6th problem set, (emphasis on Chapter 9,10). Not to turn in.

Department of Electrical and Computer Engineering

The University of Texas at Austin

EE 306, Fall, 2019
Yale Patt, Instructor
TAs: Sabee Grewal, Arjun Ramesh, Joseph Ryan, Chirag Sakhuja, Meiling Tang, Grace Zhuang

Introduction to EE 306
August 28, 2019

Class meets: 3:30 to 5pm, MW in WCH 1.120. There will be 8 discussion sections, as shown on The University's Fall Course Schedule. You are registered for one of them, and in fact, your registration is tied to the unique number of that section. The first Discussion Session will be held this Thursday or Friday, August 29 or 30. I would like you to go to the session that you are registered for this week. However, after this week, if you find it more comfortable to change sections for any reason, feel free to do so. Education works best when you and the instructor are on the same wavelength, so you are encouraged to shop around for the TA who teaches best for your needs. Problem sets will be returned in discussion section, so you will need to identify on those pieces of work which section you wish to have your problem sets returned to. Other than that, what section or sections you attend is completely up to you. Your final grade in the course will be reported by me in December, and as long as your name appears in one of the sections, it does not matter which one. Also, if you miss your discussion section, and want to attend one of the others, that is fine also. Or, if you want to regularly attend more than one section to get more than one perspective, that is also okay.

We will also have additional help sessions earlier in the week in the late afternoon or evening for students who feel they need extra help. These will not be official discussion sessions, but will be available to whoever needs them. Feel free to attend any of them if you find them useful.

We will not take attendance. You are in college now, and with that (in my view) comes the responsibility for managing your time, and deciding how best to learn what you need to know. That is, you get to decide whether or not to come to class. I should point out, however, that the TAs and I are very unsympathetic to a student who does not attend class or discussion section and then expects us to make up for that with

private tutoring. I should also add that this is not my first year of teaching, and my experience has been that if you don't come to class, you probably don't pass the course.

I am also well aware of the fact that this is for many of you your first venture away from home, and you don't have your mother telling you to do your homework while she is making your bed for you. You are on your own, and there are plenty of distractions just waiting to do you in. I am not suggesting that you abstain from all distractions and have no fun at all this semester. I am suggesting that your ability to manage fun and study will have a lot to do with how successful you are here. Too many students wait until they have flunked the first exam to recognize that things are different than they were in high school.

You are strongly encouraged to form study groups, and share your insights as well as challenge each other's mastery of the material. Mastery of the material will be very important to success in this course. We are well aware that you have gotten great mileage from your ability to memorize. Our experience is that memorization ability will not work here. The exams are geared to testing your deep understanding, and NOT your ability to memorize. Study partners can provide the necessary foil to test whether you really understand the material.

Our experience also tells us that if you form an effective study group, you will probably all do better in the course than if you go it alone. Life (and certainly life after graduation) is about working in teams, and you are encouraged to do so in EE 306.

Achievement will be based to a large extent on your performance on the three exams. Copying others' homework without it going through your head is a sure way to fall flat on your face on the exams.

This is an engineering course. It will continually build on the knowledge you have previously gained in the course. It probably is not a good idea to let things slide and try to catch up at the last minute. Please do not consider that a challenge.

Statement of the objectives of the course:
This course is a serious introduction to the fundamental underpinnings of computing. It is a **first** course. We assume you are enrolled because you want a serious introduction to computing.
Our objective is to remove a good deal of the mystery of how computers work and to teach you enough programming methodology to enable you to get the computer to do useful work for you. In that vein, we will start at the bottom and work our way up. In every case, when we cover a sophisticated topic, we will try to tie it to what you already know. We expect you to come out of this course not only knowing how to do

some things, but also having a deeper understanding of why some of those things are as they are.
We should also tell you at the outset that this course represents a major departure from the way computing has been introduced at most universities over the past 30 years. Most universities have been starting with an introduction to programming. However, there is a lot of talk among computer education professionals lately that the historical approach is not the best approach. We pioneered the approach of EE 306 for the first time at the University of Michigan in the Fall of 1995. As of this moment, about 300 colleges and universities have adopted our approach, and the number is continuing to grow. We in ECE at UT strongly agree with this approach. That is why EE 306 is now the first required computing course for all electrical and computer engineering students.

Course materials: Textbook: "From Bits and Gates to C, C++ and Beyond, 3rd edition," by Yale N. Patt and Sanjay J. Patel.
Simulator: Available on Windows, Unix, and MacOS platforms. Your TA will show you how to access it when it becomes necessary to use it. You are also encouraged to read the tutorial on the LC-3 Simulator, available from the EE 306 web page.
The programming assignments will require using the LC-3 (for Little Computer 3), a machine invented specifically for this course. The reason for the "3" is that we did not get it right the first time, ...or even the second time. You will write, debug and run each of the programs on the LC-3 Simulator.
Finally, we will post material from time to time on the web that we think you will find useful. Please check the website regularly. I would say, at least every other day. **If an announcement has been posted on the website for more than three days, and you are not aware of it, the fact that you are not aware of it will not be considered a legitimate excuse.**

What we expect from you: We do not expect to read the book to you. You should consider the lectures, the discussion sections, the book, and any material on the website as different mechanisms for helping you learn the material. We will expect you to write five programs and solve six problem sets. On the problem sets you are encouraged to work together in groups. The programming assignments you may work in groups up to the point where you begin the actual construction of the program. That is, you can work together on the high-level discussion of the problem. But once you start designing the details of the algorithm, even before you start writing code, you are on your own. The problem sets and the programming assignments are ways for you to check to see if you are getting the material. If you are not getting it, please see one of the TAs or the instructor for help.

We encourage you to study in groups, and, where practical, to come to various office hours in groups. That usually will result in all of you understanding the material better. You are encouraged to ask questions after you have thought about the material,

and to challenge assumptions. Computer Engineering deals with "nature" that is man-made (person-made, actually, but that is awkward) and so we the people may have made it wrong.

Your own work: Although we encourage you to study together, the actual programs you write and examinations you take MUST be your own work. Providing information to another student where prohibited, or obtaining information from another student where prohibited is considered cheating. This includes the exchange of any information during an examination and any code that is part of a solution to a programming assignment. Allowing another student to read something on your paper during an examination is considered cheating. In fact, leaving information unprotected so it can be compromised by another student is considered cheating. This includes sheets of paper lying about in your dorm room, and computer files that are not properly protected. If you cheat, you violate the soul of the University, which we take very seriously, and will deal with in the harshest possible way. **If you have any question as to what is permitted and what is not, ask the instructor or a TA FIRST.** If you don't ask first, and you do something that is not allowed, the response "I thought it was okay" is not an acceptable justification. I am embarrassed to have to bother all of you with this paragraph, since for most of you, the contents of this paragraph is totally unnecessary. But, every semester there are some who feel it is okay to cheat. And, to attempt to deter them from cheating, I apologize for having to take up all this time of the rest of you.

For those of you who decide to continue in this course, Good Luck. We hope you find the experience an effective initiation to your serious computing education. We also hope you have a good time during it.
