Texas State University
Part-Time Faculty Excellence in Teaching Award
Nomination Form

Name Ted Lehr Net ID t_l81

Department Computer Science College Science and Engineering

Current TXST teaching appointment FTE% 20

Number of long semesters of TXST teaching at 50% or more FTE 0

Brief statement (100 – 150 words) of why the nominee is deserving of this award:

I have taught one or two sections of fundamental computer science courses since January, 2014. My classes make the material very relevant and interesting to students. Many finish the classes with, if not a passion for the material, then a happiness that they've learned something relevant and interesting in its own right. I am advisor for our student organization, EXE with whom I have discussed how to make the club more relevant to students and to the university. To that end, the students have initiated projects to increase recruitment, provide skill set development not offered by their department, increase interest of women in computer science and launch a popular Facebook presence. Finally, I have assisted Texas State in my role as Data Architect for the City of Austin by connecting faculty from many departments to the City and major companies like IBM and AT&T for research collaboration.

What are your personal strengths as a teacher?

- I like teaching, especially the figuring out of how to get students passionate and curious. I enjoy the good days and find the bad days puzzling, like problems to be understood and solved. I like the students. I find their differences from me to be fascinating, confounding and revitalizing. They keep me "on my toes." I believe they sense my passion for the topics and my desire to have them not only understand the subject, but enjoy understanding it, including its history and the colorful personalities and twists that got us where we are.

- I have had no formal, extended time focused on teaching. The closest approximation to working with young people as a soccer coach and as father of two very fine young men. Yet, I have almost 25 years of experience in industry, large and small. Established and start up. In industry, I've held positions from executive management to research, to customer support, to QA, development and engineering management. I have also participated in municipal government as a volunteer city commissioner for a small Texas town. And now I'm currently a data architect for the City of Austin.

- I therefore have insights into many of the types of careers the students will start or evolve into as well as the careers of those will manage and evaluate them. These insights inform how and what I teach with the probable effect being that students experience more of a practical, "here's what you're going to do with this knowledge when you get out there" approach rather than one favoring theory. Is this the best approach? Well, it's what I can deliver best. And when I engage the students with a goal that they develop a love of learning, then the important theoretical aspects will come in time and with enthusiasm.

How has your teaching changed since you began teaching and what have you done to improve it?

- When I began teaching, my most pressing challenges were:

  - I didn't know what the students didn't know. I hadn't experienced what students know or how they think. How to read the students? Would I be two simplistic or too high level?

  - I hadn't put a college curriculum together before nor had I delivered it. Even as I borrowed from my generous and helpful colleagues, it wasn't clear I could execute effectively.

  - Although I am serenely comfortable speaking in front of people, it has been mostly in a leadership, management or information giving roles. Teaching had to do that as well as inspire, guide and assess.

- My student reviews have been quite good and have corroborated most of my assessment of my challenges. I've used the feedback to continue and evolve what I agree are my good points: that I'm enthusiastic, caring, and full of valuable "industry experience." But I've cherished the criticisms ... that I go too fast, that I can appear unprepared and sometimes make mistakes in class demonstrations ... to make changes to go over material more thoroughly.

- Another improvement: I'm informed by my long time in industry and management. As a manager, I expected my employees to work together to figure out the answers to technical problems themselves. As a lecturer, some might argue that I should be able to answer any relevant course question. As manager, I expected crisp, professional, friendly, confident communication. As lecturer, I've experienced youthful and timid communications and immature behaviors. That was no surprise. It turned out, however, that my skills in handling those were unpolished and tentative.

- Yet, I've turned this "manager's expectation" to the students' advantage by encouraging boldness, confidence and crisp communication. I'll say "I teach what I want to hire."   What I want to hire are smart, bold, creative and confident graduates.

Give an example of a teaching challenge you have encountered and explain how you've dealt with it.

- CS 3398: Software Engineering is listed as a "writing intensive" course. The students are assessed on their writing and other communication skills. As usually taught, this involves writing a 20 or 30 page design document by the end of the course that is used to assess the student's writing while providing a mechanism for them to learn software engineering concepts.

- Long prescriptive design documents, however, are rarely used anymore in the software industry except when required by customer contract, the best illustration being contracts with the United State Department of Defense. Instead, software engineering practices that embrace team based "Agile" or just-in-time documentation and communication are preferred in web, mobile and cloud industries where rapidly changing customer requirements mean software products must be frequently updated. Thus, the long design documents, despite their suitability for assessing a student's writing skills, do not adequately prepare the students for the most common contemporary software engineering practices

- Thus the challenge was how to teach "state of the art," contemporary, market relevant software engineering concepts and behaviors while not relying on long written documents but still assigning work products that could be used to improve and measure student writing and communication skills.

- Contemporary, team based protocols for intra-team communication rely on social media like technologies adapted to engineering problems. These include discussion bulletin boards, on-line technical documentation and reporting and visibility up-and-down an engineering organization. They demand … but don't always get …. concise, well written and well argued engineering prose.

- By requiring the students to use tools like IBM's Rational Suite and popular and well-run software forums like Stack Overflow, I was able to assess the students' writing while exposing them to contemporary software engineering technologies that work best when the participants write well, professionally and thoughtfully.

Please give examples of innovative assignments and course design components that promote active learning and/or engagement.

- As sometimes taught, CS 3398: Software Engineering, emphasizes design documentation and mastery of software engineering concepts. Although learning software engineering concepts is important, without practice in applying them the result is like being an expert on recipes without knowing how to cook. Good cooks know when to follow and when to deviate from recipes. I want the students to cook. That means they have to put the concepts into practice.

- Two innovative assignments, among others, helped the students learn to put the concepts into practice:

  - **Stand ups**: This is an exercise, when done in industry, is performed every day by a team of 5 - 10 engineers. It is not possible to do every day in a class, so we targeted once a week. Stand-ups last 5 - 10 minutes, during which the team reviews an on-line "burn down" chart that describes the team's progress during the current development cycle, typically 2 - 4 weeks. Each team member describes what s/he did since the previous stand up, what s/he intends to do by the next one, and what "blockers" exist that are inhibiting work. The exercise encourages team work, early communication, accountability to each other and immediate feedback on the status of development objectives.

  - **Posting to discussion forums**: One of the resources software engineers need to be comfortable with is on-line engineering forums. The students were required to write questions and, instead of asking me for answers, post them to one of two forums : Stack Overflow or IBM's RTC Forum. They were judged on the professionalism, thoughtfulness, writing quality and relevance of the question as well as the quality of the answers. Experience using and writing to these forums is, today, a skill on par knowing how to use a search engine well.

Three Written Comments from Students

1. The most relevant and useful computer science course I have taken at Texas State. – Fall -2015, CS 3398 – Software Engineering student evaluation.
2. When an instructor cares about what they are doing, you can tell. Dr. Lehr obviously cares. Fall-2015, CS 3398 – Software Engineering student evaluation.
3. This is the best professor I have ever had. He made an amazing effort to align his teaching with students' learning preference. Listened to the students' recommendation and put in all means to make the course fun and interactive. - Spring-2014, CS 2420 - Digital Logic student evaluation

Course Syllabus

See saved PDF.

CV

See save PDF

TEXAS STATE VITA

Name: Ted Lehr **Title**: Lecturer

Educational Background

Ph.D., 1990, *Carnegie-Mellon University*, Pittsburgh, PA, Electrical and Computer Engineering, "Compensating for Perturbation of Asynchronous Computations by Software Performance Monitors."

MS, 1985, *Carnegie Mellon University*, Pittsburgh, PA, Electrical and Computer Engineering, "Implementation of a Production System Machine."

BSBA, 1983, *Bucknell University*, Lewisburg, PA, Electrical Engineering and Philosophy

Professional Experience

Position	**Entity**	**Dates**
IT Data Architect	City of Austin, Texas: Communications and Technology Mgnt	2014 - present
Senior Software/Research/Support Engineer	CA Technologies (acquired Hyperformix),	2009 – 2013
Sr Project Manager	Hyperformix	2006 – 2009
Vice President, Development	Hyperformix/SES	1999 – 2006
Intel Project Manager	SES	1997 – 1999
Advisory Software Engineer	IBM	1990 - 1997
Intern	Honeywell Computer Sciences Center	1985
Intern	Sperry Corporation	1984

Edited Books

1. Ted Lehr, Z. Segall, D. Vrsalovic, E. Caplan, A.L.Chung, C.E. Fineman, "Visualizing Performance Debugging," *IEEE Computer,* October, 1989
   - Reprinted in *Datapro Management of Application Software* by Datapro Research, McGraw-Hill Information Services Company, 1990

2. Theodore F. Lehr, Bob Wedig, "Toward a GaAs Realization of a Production-System Machine," *IEEE Computer*, April, 1987
   - Reprinted in *Reduced Instruction Set Computers* by W. Stallings, IEEE Computer Society Press, 1990
   - Reprinted in *Gallium Arsenide Computer Design* by V.M.Milutinovic, D.A. Fura, IEEE Computer Society Press, 1988

Refereed Journal Articles

1. Ted Lehr, Z. Segall, D. Vrsalovic, E. Caplan, A.L.Chung, C.E. Fineman, “Visualizing Performance Debugging,” *IEEE Computer,* October, 1989
2. Theodore F. Lehr, Bob Wedig, “Toward a GaAs Realization of a Production-System Machine,” *IEEE Computer*, April, 1987
3. Ted Lehr, “Development of a Low Cost Graphics Terminal,” *Journal of the Computers in Education Division of ASEE*, January – March, 1985

Refereed Conference Proceedings

1. Leslie Martinich, Ted Lehr, Deepika Sangam, “Make the World a Better Place: An Association-Industry-Academia Partnership,” International Stem Education Conference, April, 2014.
2. T. Lehr, M. Breternitz, Jr., A. Gheith, A. Jindal, J. Peterson, J. Van Fleet, “Adapting AIX to a Shared Memory Cluster,” *Proceedings of the Share Europe Anniversary Meeting, October*, 1993, p 415.
3. Anita Jindal, Ted Lehr, “Measuring and Visualizing Client/Server Behavior,” *Proceedings of the Share Europe Anniversary Meeting*, October, 1993, p. 429
4. Ted Lehr, John Florkowski, “A Visual Comparison of AIX and OSF Using PIE,” *1992 IBM Performance ITL*, Toronto, Ontario, April, 1992
5. Ted Lehr, David Black, Zary Segall, Dalibor Vrsalovic, “Visualizing System Behavior,” *1991 International Conference on Parallel Processing*, August, 1991.
6. Ted Lehr, “Visual Performance Monitoring,” Invited Talk, Polish Institute of Technology, Warsaw, Poland, November, 1990
7. Ted Lehr, David Black, Zary Segall, Dalibor Vrsalovic, “Visualizing Context-Switches of Parallel Programs Using PIE and the Mach Kernel Monitor,” *1990 International Conference on Parallel Processing*, August, 1990
8. D. Vrsalovic, Z. Segall, D. Siewiorek, F. Gregoretti, E. Caplan, C. Fineman, S. Kravitz, T. Lehr, M. Russinovich, “Performance Efficient Parallel Programing in MPC,” *22nd Hawaii International*
9. Theodore F. Lehr, Bob Wedig, “The Implementation of a Production System Machine,” *19th Hawaii International Conference on System Sciences*, January, 1986

Reports

1. Ted Lehr, Bret Olszewski, Robert Berry, “Improving Performance of Visual trace Analysis Using Abstract Time-Lines*,” IBM Technical Disclosure Bulletin*, March, 1995.
2. Ted Lehr, “Improving Performance of Visual Trace Analysis Using Abstract Time-Lines*,” IBM Technical Disclosure Bulletin*, March, 1995.
3. Ted Lehr, “Improving Performance of Trace Analysis Using Visual Editing*,” IBM Technical Disclosure Bulletin*, March, 1995.
4. Ted Lehr, “Fast and Automatic Identification of Performance Outliers in Trace Data*,” IBM Technical Disclosure Bulletin*, February, 1995.
5. Ted Lehr, “Maintaining Magnification (ZOOM) Histories for Trace Visualization Tools*,” IBM Technical Disclosure Bulletin*, February, 1995.
6. Ted Lehr, “Improvement of Insertion Algorithm for Visualization*,” IBM Technical Disclosure Bulletin*, June, 1994.

7. Ted Lehr, “Run-Time Customizing of AIX Trace Visualization Tools*,” IBM Technical Disclosure Bulletin*, May, 1994.
8. Ted Lehr, “A Data Structure and Insertion Algorithm for Representing Asynchronous Occurrences for Visualization by Trace Visualization Tools,” *IBM Technical Disclosure Bulletin*, July, 1993.
9. Ted Lehr, David Black, Zary Segall, Dalibor Vrsalovic, “MKM: Mach Kernel Monitor: Description, Examples and Measurements,” *Technical Report CMU-CS-89-131*, Department of Electrical and Computer Engineering and the School of Computer Science, Carnegie Mellon University, April, 1989

Patents

1. U S Patent 7,734,775 : “Method of semi-automatic data collection, data analysis, and model generation for the performance analysis of enterprise applications,” (collecting data on applications and automating the creation of models to predict their performance) Ted Lehr and others, June 2010.

2. US Patent 7,596,546, “Method and apparatus for organizing, visualizing and using measured or modeled system statistics,” (useful ways to capture and present system performance information) Ted Lehr and others, Hyperformix, September 2009.

3. US Patent 7,290,048, “Method of semi-automatic data collection, data analysis, and model generation for the performance analysis of enterprise applications,” (useful ways to capture and present application performance information) Theodore Lehr and others, Hyperformix, October 2007.

4. US Patent 6,741,869, “Radio-like appliance for receiving information from the internet,” (an internet appliance for listening to radio signals) Theodore F. Lehr, IBM Corp. May, 2004

5. US Patent 5,898,873, “System and Method for Visualizing System Operation Trace Chronologies,” (a Ghant chart like representation of inter-related system traces) Theodore F. Lehr, IBM Corp, 1999

6. US Patent 5,777,622, “Method for Identifying Invisible Objects Ordered in a Tree-like Data Structure By Finding Adjoining Hidden Objects” (method for indicating to users that objects are in the view that are too small to display at the current resolution) Theodore F. Lehr, IBM Corp, 1998

7. US Patent 5,649,085 “Method and system for storing and displaying system operation traces with asynchronous event pairs,” (method for displaying asynchronous activity spawned by synchronous activity) Theodore F. Lehr, IBM Corp, July, 1997

Patents Pending

1. CA Technologies patent filing: CA20131009 063170.9903: “System and Method for Filtering Performance Metrics” (treating computer performance metrics as signals), Ted Lehr and others. July, 2013

2. CA Technologies patent filing: CA20130235 MBSS Ref: 1100-130235: “Methods, Systems, and Computer Program Products for User Side Optimization of Acquisition of Virtual Resources”, Ted Lehr and others, July, 2013.

Invited Talks, Lectures, and Presentations

1. 3-4/2015: Lectures, panel discussions at University of Texas, Ischool on data in public service and public service career opportunities around data. Guest lecture in journalism and sustainability classes at Texas State University on the role of data in the respective discplines.
2. 1/2015: Facilitated research collaboration discussion between City of Austin, Transportation Department, the Texas Department of Transportation and Texas State and University of Texas researchers to improve fire emergency services support in Austin through an NSF research grant.
3. 1/2015: Presented research and other open data opportunities to Open Austin, a civic technology organization committed to improving the community through technology.
4. 11/2014: League of Cities Conference, Austin, Texas. Hosted government officials from around the United States at a workshop at IBM-Austin to discuss the role of open data and open data policy in "smarter" cities.
5. Ted Lehr, Automating the Modeling of the Scalability of Computing Systems Performance Using Industry Standard (SPEC) Benchmarks, IEEE Central Texas Consultants Network, September 2012
6. Ted Lehr, Applying predictive analysis and algorithms to prevent capacity problems in moving to or operating under virtualized cloud environments, IEEE Computer Society, Austin Chapter, April 2012
7. Ted Lehr, Zary Segall, "Measuring Performance and Understanding the Numbers You Get," Invited Talk, Information Technology Center, Carnegie Mellon University, Pittsburgh, PA, March, 1991
8. Ted Lehr, "Visual Performance Monitoring," Invited Talk, Warsaw Institute of Technology, Warsaw, Poland, November, 1990

Submitted, but not Funded, External Grants and Contracts

- January, 2015, As City of Austin Data Architect, facilitated submission of NSF grant proposal: Partnerships for Innovation: Building Innovation Capacity – program solicitation NSF 14-610. Proposal led by Texas State, with partners from the City of Austin, IBM, ATT, Texas Department of Transportation and the University of Texas.

Funded, External Grants and Contracts

IEEE Foundation Grant #2014-039FF: "'Make' the World a Better Place," 8/2014 – 6/2015: $21,450.

Fellowships, Awards, Honors

- 1990 - 1997: Several technical achievement and innovation awards, IBM - Austin
- 1985: Burroughs Graduate Fellowship, Carnegie Mellon University
- 1983: General Electric Graduate Fellowship, Bucknell engineering faculty for graduate study at the university of choice
- 1983: Phi Beta Kappa

- 1981: Tau Beta Pi
    - President, Bucknell University Chapter: 1982 - 1983
- 1979: Kodak Scholarship for outstanding freshman engineer, Bucknell University

Institutional Service

- 10/2014 – Project director, ICPC- ACM Programming Contest at Texas State University
- 9/2014: ABET Evaluator, Pennsylvania State University – Erie: Software Engineering
- 3/2014 - present: EXE Advisor: Student Computer Science Club, Texas State University
- 2014 - present: Vice-Chair (elect) IEEE Education Society – Central Texas

Community Service

- **City government service**
    - 2014 – Present: City of Austin, IT Data Architect. Improve the availability, quality and use of city's open data.
    - 2005 – Present: Chairman, City of Dripping Springs Historical Preservation Commission
    - 2006 – 2013: Commissioner, City of Dripping Springs Planning and Zoning Commission
- **Secondary school service**
    - 4/2014 – Recruited and facilitated panel discussion at Dripping Springs high school on careers for women in STEM fields.
    - 2013 – Member of the advisory committee for the Dripping Springs high school engineering academy
    - 2013, 2007, 2005: Member of Dripping Springs ISD Long Range Planning Committee
    - 2004, 2007: Chairman, "Friends of Dripping Springs Education" PAC. Led the promotion and passage of DSISD bond elections for a new elementary school and major upgrades to the middle and high schools

CS 2420 Digital Logic – Section 253 – Spring 2015 – Syllabus

Lectures:	MW 2:00 – 3:20, DERR 240
Lab:	Various, DERR 236

Instructor:	Ted Lehr
Office:	CMAL 301G
Phone:	245.3666
Email:	t_l81@txstate.edu
Office hours:	MW 4 – 6:30 or by appointment

1 Course Overview

This course covers the fundamentals of computer design by studying the algebraic methods to design digital circuits and how they can be used to design the basic blocks that are used to design a computer. An introduction to computer hardware and the technologies used to create, capture, and communicate digital information. A laboratory provides hands-on experience with the subject matter, e.g., electricity, combinational and sequential digital circuits, VLSI, etc.

3 Textbooks and websites

You may find the following books and websites helpful. There will be no problems taken from the textbook. There will be recommended reading from it, but the course material can be learned using the lecture, labs and numerous web links that will be provided.

Randy H. Katz, Gaetano Borriello, *Contemporary Logic Design* , second edition, Pearson Prentice Hall, 2005

A small sample of the web links to be referenced:
Basic Electrical Components
Boolean_algebra
Digital_electronics
Basic Logic Gates

4 Prerequisites

CS 2318: Assembly Language or concurrent enrollment in this course and Discrete Mathematics I

5 Grading Policy

Homeworks and Quizzes − 30%

Midterm − 20%
Final − 25%
Lab − 25%
Lab:
You must attend a designated lab (Counts 25% of your grade). For details on the labs, go to the department web site http://labs.cs.txstate.edu/2420/ for more information, as the lab schedules begin.

- Homework: One homework (about) every two weeks.
- Quizzes: Quizzes to be unannounced.
- Exams: *Approximate* dates
    - Mid-Term: Wednesday, March 11, 2:00 – 3:20 pm
    - Final: Monday, May 4 or 11, 2:00 – 4:30 pm
- Both mid-term and final will open-notes: one letter size paper (no electronic devices allowed).

6 Course Overview

This course will cover the following areas in detail:

- Understand basic Electrical Engineering (EE) concepts relevant to digital logic including Ohm's Law, resistors, and transistors, FET technology and how they are used to design computer circuits.
- Learn about number systems, conversions between them and how they affect how we count and how they are used in computer arithmetic.
- Boolean Algebra and its use in computer design, including fundamental identities, universal sets, DeMorgan's Theorem and logic minimization.
- Combinational Logic Design including analysis and synthesis of logic circuits
- Elementary concepts in Computer-Aided Design, specifically, automatic synthesis
- and simulation
- Combinational Logic Blocks, e.g., Adders, Multipliers, Decoders, etc
- Sequential Logic Design, including Moore and Mealy sequential circuit models, state assignment and minimization
- Sequential Logic Blocks, e.g., Flip-flops, Registers, Memories, Counters, etc
- Computer Components, including Processing units and their components, I/O devices, etc.

7 Learning Objectives

- Basic understanding of electricity, electronics and CMOS technology as they pertain to the logic implemented by digital circuits.
- Introduction to digital logic design: logic and gates, combinational logic design, sequential logic design, FSMs, registers (counters and shift registers), PLA-type devices, what clock does, basic computer components and basic computer architecture

- Communications concept: parallel & serial transmission, asynchronous and synchronous communication, rudiments of information theory.

8 Course Policy

Attendance: Class attendance, where most of the subject material will be discussed and homework assignments reviewed, is essential to success. For this reason, attendance is strongly recommended. Students are fully responsible for all material presented, assigned in class, and responsible for being in class for all tests. You must read your emails fairly often to keep up with class announcements.

Homeworks, Quizzes and Exams: 10% of the assignment score will be based on clarity of exposition of the solutions. In other words, a problem solution that is technically perfect, but which is presented in a difficult-to-understand manner, will lose 10% of the score for that problem. *Neat and concise solutions are required in order to receive full credit for your solutions.* If you cannot solve a particular problem, state this clearly in your write-up, and write down only what you know to be correct; rambling at length about ideas that don't quite work may cause additional points to be deducted.

Late Policy: Without prior arrangements, missed exams, homework and projects result in a grade of zero. In order to take a make-up exam, contact me (Dr. Ted Lehr). Prior to the exam if you have to miss it for some valid reason; documentation may be required. Notification after the exam will result in a score of zero. There are no late days of homework. Late homework s ubmissions will not be graded for credit. In extra-ordinary cases, contact me (Dr. Ted Lehr) prior to the due date to get homework and project extensions.

9 Academic Integrity: Cooperation vs. Cheating

Working with others on assignments is a good way to learn the material and I encourage it. However, there are limits to the degree of cooperation that I will permit.

When working on assignments, you must work only with others whose understanding of the material is approximately equal to yours. In this situation, working together to find a good approach for solving a problem is cooperation; listening while someone dictates a solution is cheating. You must limit collaboration to a high-level discussion of solution strategies, and stop short of actually writing down a group answer. Anything that you hand in must be written in your own words. If you base your solution on any other written solution (copying), you are cheating.

When taking a quiz, you must work completely independently of everyone else. Any collaboration here, of course, is cheating.

I do not distinguish between cheaters who copy others' work and cheaters who allow their work to be copied.

The punishment for cheating on an assignment or quiz will be the docking of the final grade by one mark (so, a C instead of a B for example). If two people are caught sharing solutions then both the copier and copiee will be held equally responsible. Cheating on an exam will result in failing the course.

See Student handbook for more information about Texas State Academic Policies including probation, suspension, academic honesty, dropping aclass, incompletes, grade change, and withdrawal.

10 Students With Disabilities/Special Needs:

If you are a student with a disability who will require an accommodation(s) to participate in this course, please contact me within the first two weeks of the semester. You will be asked to provide documentation from the Office of Disability Services. Failure to contact me in a timely manner may delay your accommodations

11 Tentative Calendar of Lecture Topics

This calendar and the order of topics may change but it is a reasonable approximation of what we will cover.

Week	Date	Topic	Lab
1	1/21/15	Syllabus and Introduction to Circuits	
2	1/26/15 1/28/15	Numbers	
3	2/2/15 2/4/15	Gate Algebra	Lab1: Introduction to digital logic circuits and instruments
4	2/9/15 2/11/15	Gate Algebra	Lab2: Transistors and logic gates
5	2/16/15 2/18/15	K-Map terms	Lab3: Circuit performance
6	2/23/15 2/25/15	K-Map adder	Lab4: LogicWorks
7	3/2/15 3/5/15	Combinational Logic Analysis	Lab5: BCD seven segment
8	3/9/15 3/11/15	Midterm Review and Midterm	Project: Part 1- Combinational logic component
9	3/16/15 3/18/15	Spring Break	Lab6: Latch and flipflop
10	3/23/15 3/25/15	Finite State Machines: Mealy	Lab7: Mealy FSM
11	3/30/15 4/1/15	Finite State Machines: Moore and Registers	Lab8: Moore FSM

12	4/6/15 4/8/15	Register Counters and Counter Memory	Project: Part 2- Sequential logic component
13	3/13/15 4/15/15	Memory and register-transfer-logic	Lab9: Memory
14	3/20/15 4/22/15	Intro to Computer Design	Lab MakeUp
15	3/27/15 4/29/15	Computer Design and Final Review	
16	5/4,6 or 11/15	Final Exam	