COURSE SYLLABUS

**********YE

AR COURSE OFFERED: 2016

SEMESTER COURSE OFFERED: Spring

DEPARTMENT: Computer Science

COURSE NUMBER: CS3432

NAME OF COURSE: Computer Architecture I

NAME OF INSTRUCTOR: Dr. Eric Freudenthal

**********T

The information contained in this class syllabus is subject to change without notice. Students are expected to be aware of any additional course policies presented by the instructor during the course.

****

Learning Objectives

Learning Outcomes for CS3432, Computer Organization (Catalog Title: Architecture 1)

family	Prerequisite knowledge from CS1,CS2, digital design, discrete, and precalc	Students will be familiar with...	Students will be able to effectively apply skills...	Students will be able to analyze & synthesize solutions..
numeric representation & ops	- familiar with radixes, signed representations, and scientific notation		- convert hex, dec, signed-dec, binary binary metric - signed/unsigned comparison (flags,order) - Add-with-carry - cast/sign-extend - floating point	- can determine appropriate representations for elementary types and design low-level programs that compute arithmetic results

COURSE SYLLABUS

linearization	- algebra, - arithmetic & control-flow structures of an oo language - block structures,		- expressions (incl side effects) - control flow (if/while/for) - translate boolean logic - branch tables - op on arrays, structs, and pointers	- can translate infix expressions and block-structured programming constructs to assembly language
gross architecture	- able to program in at least one oo language - familiar with cobinational and sequential logic	Can describe the fetch-execute cycle in the context of the roles of PC, SP, flags, registers and memory.	- select appropriate instructions - specify operand order - encode and decode instructions - specify addressing mode - utilize interrupt mechanism - implement interrupt handlers	- can implement and debug simple imperative programs in assembly or machine language
timing	- algebra - synchronous logic		- determine cycles/instruction - determine which instructions repeat in a loop	- can compute the execution time of a simple loop
subroutine linkage & separate compilation	- in oo languages studied in CS1/2		- parameter passing - return value - allocation of auto vars - register usage - global/local symbols	- can write or call a method with local variables, parameters, and return value in assy lang
variable allocation	- in oo languages studied in CS1/2		Can define and use variables with various.. - scope: visibility (file/method/program) - variable lifetime (program/method) - size - alignment - arrays - pointers - structs	Can appropriately allocate static and auto variables including arrays and pointers in assembly language

COURSE SYLLABUS

tools	- ide - hierarchical filesystems		Can effectively employ in the composition and debugging of programs - editor - compiler - make - svn - bash - gdb	Can compose and debug simple programs in a command-line environment
written communica-tion	- proficient in english		Can - interpret technical documentation on familiar topics - describe implementations that they design - recognize/use technical terminology - appropriate documentation for code	Can appropriately document simple programs
mature programm-ing	- proficient in OO programming - appropriate comments - can modularize - appropriate symbol names - coding style		Can utilize in a program - appropriate comments - modularize - imperative programming - appropriate symbol names - coding styles	Can appropriately modularize and document simple programs consisting of multiple files
advanced topics		- pipelining - vectoriza-tion - predicated instructions		Can identify when these topics are relevant to constructing an efficient solution.

devices	gates, latches, (de)multiplexers, ALUs, switches, counters	- gross characteriscs of memory & storage devices - counter-timer	can implement - simple programmed i/o - interrupt handlers	Can design programs that implement simple programmed i/o and interrupt handling - can determine the types of storage devices suitable for a variety of uses.

Students are also expected to demonstrate two generic families of skills (which are also assessed).

- Clerical accuracy: clerical errors do not significantly interfere with students' ability to construct correct solutions.
- Completion: students are able to construct complete solutions in the allotted time.

A weekly schedule of course topics is published on the course web site.

Labs

In addition to the "lecture" course section, students must also register for and attend a separate lab section. All lab assignments must be completed and submitted on time as specified by the course web site except in exceptional circumstances explicitly approved by the instructor.

Course Grading

Mid-term and final course grades are equal to the fraction of course learning outcomes for which the student has demonstrated mastery via solutions to problems within assessment instruments (quizzes, and exams, and labs).

Full credit for a skill is assigned if mastery is demonstrated three of the last five times it is assessed within a quiz, exam, or lab assignment. Letter grades correspond to the fraction of skills (course learning outcomes) for which mastery is demonstrated using the conventional 100 point percentage scale:

A: > 90%; B: >80%, C: >70%, D: >60%

Mastery is assessed using the following scale:

- Insufficient proficiency: no credit
  Solutions do not indicate mastery as indicated in learning outcomes.

    - 0: no evidence of familiarity
    - 1: familiar, but not proficient
- Sufficient proficiency: full credit
  Solutions indicate mastery as indicated in learning outcomes.
    - 2: proficient: Solutions indicate ability to apply skill in problem-solving.
    - 3: exceptional: Solutions indicate exceptional mastery.

Students are provided multiple opportunities to demonstrate most skills during the course, and most problems provide opportunities to demonstrate multiple skills.

Quiz/Exam Schedule:

- Final exam: date and time are specified by the university.
- Quarterly tests: dates will be announced in class and indicated on course web site.
- Frequent Quizzes
    - Generally unannounced
    - At least one every two weeks.
    - Generally do not need to be "made up"

Required Reading

- Absolutely required: Kerningham, Brian W & Ritchie, Dennis M. "The C Programming Language, Second edition," Prentice Hall, ISBN: 0-13-115817-1.
- The course web site contains an online text on assembly language programming for the MSP-430.

Recommended Reading

- MSPGCC cross-tools manual (55 pages). Can be downloaded from the course web site..
- Recommended by students (and by no way required): Android app "Programmer Mental Math" by Joel Jurix.

STANDARDS OF CONDUCT:

Students are expected to conduct themselves in a professional and courteous manner, as prescribed by the Standards of Conduct. Students may discuss programming exercises in a general way with other students, but the solutions must be done independently. Similarly, groups may discuss project assignments with other groups, but the solutions must reflect their own creative work. Graded work should be unmistakably your own, and (portions of) solutions developed by others must be clearly documented. Transcriptions and paraphrasing of others' solutions are strictly prohibited. Professors are required to report academic dishonesty and any other violation of the Standards of Conduct to the Dean of Students.

COURSE SYLLABUS

Disabilities and Special Circumstances:

All students should have the opportunity to succeed. It is essential to determine appropriate accommodations when a disability or personal/life circumstance has the potential to significantly interfere with your ability to succeed in this course.

- Disabilities, other significant circumstances, and accommodations must be documented and by Center for Accommodations and Support Services (www.utep.edu/cass).
- Students are encouraged to discuss course-specific needs and accommodations with the course and lab instructors.