

CS-3432 Computer Architecture I

Contents

- [1 Course Info and Lab Notes](#)
- [2 Homework, tests, and labs](#)
- [3 Readings](#)
- [4 Resources](#)
- [5 Disabilities and Accommodations](#)
- [6 Academic Honesty](#)

Course Info and Lab Notes

- Administrivia
 - Instructor: [Dr. Eric Freudenthal](#) (see homepage for office hours)

TA office hours:

- TAs will be present in CCSB 3.1202 for the first 10 minutes of their office hours.
- If nobody is present or expected, they will leave.
- If you need to meet a TA at another time, send them email.

TA Schedule	David Pruitt	Daniel Cervantes	Adrian Veliz
Arch lab		MW 1:30-2:50 TR 12:00-1:20	MW 9:00-10:20
Office hours	R 12:00	TWR 4:00-5:00	

- Join the course email listserv using your google (gmail) identity: <https://groups.google.com/forum/#!forum/2016-fall-arch1>
- [Course Outline](#): topic sequence including **lecture notes**
- **Prerequisites**:
 - To succeed in this course, you need to have analytic maturity with concepts taught in digital design (EE2369/2169), elementary data structures and algorithms (CS2401), and discrete math (Math 2300).
 - As stipulated in the course catalog, the usual way to demonstrating this maturity is by earning a C in (1) all of these courses and (2) Data Structures (CS2302).
 - I'll waive the prerequisite requirement for CS2302 for students who earn at least B's in EE 2369/2169, CS 2401, and Math 2300.
- **Labs**
 - Students **must** enroll in a lab section (typically meets MW or TR)
 - **Participation is mandatory.** It is extremely easy to fall behind and imperative that you make arrangements with the instructors or TAs to make-up the missed lessons and work. Lab sessions and assignments managed by TA. Students at risk of failing due to not engaging with the lab course may be dropped.
 - There may be scheduled or unscheduled quizzes in lab course

- Assignments and due dates: Click on [Lab Projects](#) navigation tab.
- **Texts**
 - Absolutely required: Kerningham, Brian W & Ritchie, Dennis M. "The C Programming Language, Second edition," Prentice Hall, ISBN: 0-13-115817-1.
 - MSPGCC cross-tools manual (55 pages): [pdf](#), [web](#)
 - Recommended by students (and by no way required): Android app "Programmer Mental Math" by Joel Jurix.
- Course leaning outcomes
 - Are enumerated in the [syllabus](#)
 - Are elaborated in greater detail in the [Grading and Course Learning Outcomes](#) page of this website.
- Attendance
 - Mandatory at all all sessions of lecture & lab course unless special circumstances arranged ahead of time with instructor

Homework, tests, and labs

- Homework
 - Students are expected to review techniques taught in class and be able to demonstrate those skills during the next class session. Example problems with solutions are published on the course web site.
- Exams/quizzes:
 - Quizzes
 - Quizzes assess individual students' abilities to demonstrate knowledge, to design solutions to realistic problems, and to present these solutions in a clear and professional fashion. Quizzes will be graded "by skill" (see below), can cover any concept or skill previously taught in the course, are generally offered at the rate of once per week, and unannounced (so students must be continuously prepared).
 - Final Exam
 - The final exam date is scheduled by the university based upon lecture time. Like quizzes, the final exam will be graded "by skill" (see below) and can cover any concept or skill previously taught in the course.
 - Rules
 - Solutions must be prepared individually without communication with or assistance from anybody except the instructor or proctor.
- Labs
 - Intention:
 - Labs are intended to provide an opportunity for students to practice and explore the use of tools and apply concepts presented in class within the context of programming projects.
 - This semester, the lab course will principally be organized around a few open-ended projects.
 - Projects will be graded during in-person demonstrations with individual students.
 - Demonstrations must occur prior to the project due date. Exceptions must be justified and pre-arranged with the instructor.
 - Students may demonstrate labs multiple times (for updated grades) prior to the project due date.
 - Demonstrations may require students to modify their programs and demonstrate competency with development tools.

- Students are expected to act professionally
 - By reading whatever resources they find relevant
 - By attributing credit to any person or reference materials that substantively contributed to their solutions
 - By only submitting solutions they fully understand.
 - Professionalism includes honesty, clarity, and accuracy.
 - Students are encouraged to help each other develop and tune their projects. There is no penalty for collaboration as long as it's documented.
- Requirements
 - **Functional:** Assignments will either require students to create complete programs or modify programs provided by the instructor.
 - **Documentation:** Submissions should include documentation that facilitates the grader's determination of
 - How to compile, use, and test
 - Principles of operation (e.g. comments & other descriptive prose)
 - Elements of the submission that were developed by others. This includes both algorithms and code. Vague attributions of credit (e.g. "Assistance was received from Jim Smith.") are unacceptable.
 - **Completeness:** Students who labs that do not substantially satisfy functional and documentation requirements will receive no credit.
- Homework:
 - Most class sessions will conclude with an assignment due at the beginning of the next class session (unless otherwise indicated).
 - While most assignments are neither collected nor graded, student mastery of relevant skills will be tested within quizzes and tests.
- **More notes on grading policy**

Readings

- Students should read chapters 1-6 of K&R (Kernigham & Ritchie's "The C Programming Language" each of the first 6 weeks of class.
- Rationale: These chapters of K&R introduce concepts in the context of C that the course examines at a lower level.

Resources

- **Obtaining and installing the Arch1 Virtual Machine**
- **MSP-430** (the embedded controller we will study)
- **Getting started in Linux/Unix and SVN**

Disabilities and Accommodations

- If you have a disability and need classroom accommodations, please contact The Center for Accommodations and Support Services (CASS) at 747-5148, or by email to cass@utep.edu, or visit their office located in UTEP Union East, Room 106. For additional information, please visit the CASS website at www.sa.utep.edu/cass.
- Students are expected to conduct themselves in a professional and courteous manner, as prescribed by the Standards of

Conduct: http://hoop.utep.edu/Student_Affairs_Chapter_One-HOP.htm Graded work should be unmistakably your own. You may not transcribe or copy a solution taken from another person, book, or other source (e.g., a web page). Copying other's work will not be tolerated. Professors are required to report academic dishonesty and any other violation of the Standards of Conduct to the Dean of Students

Academic Honesty

- Permitted collaboration: Students may discuss requirements, background information, test sets, solution strategies, and the output of their programs. However, implementations and documentation must be their own creative work. Students are required to document advice received from others and all resources utilized in the preparation of their assignments
 - **All solutions to homework and lab assignments should clearly include the following phrase: “This assignment was prepared in a manner consistent with the instructor’s requirements. Clear documentation must be provided for all significant collaboration or guidance from external sources (including algorithms or code presented in the textbook or this web site).**
 - It is not OK to submit verbatim or paraphrased copies others’ solutions or published/web resources.
- If academic dishonesty is suspected: The [student conduct office](#) will be contacted for adjudication. A temporary "incomplete" grade will be issued if their investigation extends beyond the grading period,

Commentary:

It is permissible to examine and prepare a description of a solution discovered by someone else provided that

- The “author” receiving credit has personally created submitted embodiment of a solution and analyzed their it for correctness
- The grader is provided sufficient guidance to understand aspects of the solutions attributable to others.

Appropriate collaboration is much like appropriate citation of external sources. You must indicate that your collaboration was consistent with the assignment’s requirements (studied together and individually prepared your answer). Studying together can include discussion, explanation, and (comparative) analysis of solution strategies, but can’t include anything approaching verbatim copying or rephrasing of another’s creative work. Remember that your primary goals should be:

- Helping each other identify and correct misconceptions
- Helping each other review concepts (normally called “studying together”)
- Helping each other to learn how to analytically solve problems
- Helping each other learn to communicate clearly
- Helping each other to be ready to earn a strong grade on the final exam and succeed in careers.

In all cases, “prepared in collaboration with...” is insufficient description. Better might be:

- __, ____, and I brainstormed and evaluated solutions together for problems 1,3, and 4.
- ____ proof-read my solution to problem 8 and provided some suggestions on how to communicate my ideas more clearly.
- ____ suggested a solution strategy for problem 2 and coached me in understanding why it worked.
- ____ explained how to solve problem 3 and why that solution is correct. I repeated their analysis while preparing mu solutions.

You are welcome to employ published or unpublished resources as well, providing that you properly (and similarly) categorize their contribution to your solution.