

## CS 3432 Computer Organization

Spring 2024 Syllabus

Lecture: CRN 23061; MW 9:00-10:20am; Psychology 308

Lab: CRN 24270; MW 10:30-11:50am; CCSB 1.0704

Instructor: Shirley Moore

Course webpage: <http://svmoore.pbworks.com/>

CS 3432 is a first systems course about how computer systems work. Whereas computer architecture describes the functional behavior of a computer, computer organization describes the structural relationships. CS 3432 is a course in computer organization that answers questions such as the following:

1. How are programs written in a high-level language, such as C or Java, translated into the language of the hardware, and how does the hardware execute the resulting program?
2. What is the interface between the software and the hardware, and how does software instruct the hardware to perform the necessary functions?
3. What determines the performance of a program, and how can hardware designers or programmers improve the performance?
4. What are the reasons for and the consequences of the shift from sequential to parallel processing?

### ▪ Teaching Team

- Instructor: Shirley Moore, [svmoore@utep.edu](mailto:svmoore@utep.edu)
  - Office: CCSB 3.0608
  - Office phone: 915-747-5054
  - Office hours: MW 1:30-2:20pm, others by appointment (in person or virtual)
  - Webpage: <http://www.cs.utep.edu/svmoore>
- Teaching Assistant: Robert Alvarez, [rjalvarez@miners.utep.edu](mailto:rjalvarez@miners.utep.edu)
- Instructional Assistant: Victor Lopez, [valopez9@miners.utep.edu](mailto:valopez9@miners.utep.edu)

### ▪ Prerequisites

- To succeed in this course, you need familiarity and maturity with concepts and techniques taught in digital design (EE 2369/2169), data structures and algorithms (CS 2401, CS2302), and discrete math (Math 2300).
- As stipulated in the course catalog, the usual way to demonstrate this maturity is by earning a C or above in all of the above courses.
- Students who earn B's or above in EE 2369/2169, CS 2401, and Math 2300 and are taking CS 2302 concurrently, are also considered ready.

### ▪ Lab Section

- Students **must** enroll in the associated lab section.
- Lab sessions and assignments will be managed by the TA.
- **Participation is mandatory.** It is extremely easy to fall behind and imperative that you make arrangements with the instructor or TA to make-up missed lessons and work. Students at risk of failing due to not engaging with the lab section may be dropped.
- There will be scheduled and unscheduled quizzes in the lab section.

### ▪ Texts and Readings

- Required: David A. Patterson and John L. Hennessy. Computer Organization and Design RISC-V Edition: The Hardware Software Interface, Second edition, Morgan Kaufmann, ISBN: 978-0128203316.
- Required: Kernighan, Brian W & Ritchie, Dennis M. The C Programming Language, Second edition, Prentice Hall, ISBN: 0-13-115817-1.
- Other readings and resources will be posted on the course webpage.
- **Learning Outcomes**
  - Learning outcomes are in the appendix at the end of this document.
- **Computers**
  - You will need access to a computer capable of running Linux (either natively or using virtualization software) to participate in class activities. You will need to install an ANSI C compiler before the second week of class. If you do not have access to an appropriate machine, the department has a limited number of portable computers that can be loaned to students.
- **Attendance**
  - Attendance is mandatory at all lecture and lab sessions unless special circumstances are arranged ahead of time with the instructor, or as soon afterwards as possible in the event of an emergency. Lecture sessions will be recorded, but viewing the recorded session is not a substitute for attending class. Breakout groups and hands-on exercises will be part of lecture sessions and it will not be possible to records those portions.
- **Grading Breakdown**

1000-900 = A    899-800 = B    799-700 = C    699-600 = D    599 and Below = F

  - Lab assignments and lab quizzes: 40% (400 points)
  - Class participation and Homework: 10% (100 points)
  - Quizzes: 10% (100 points)
  - Midterm exam: 15% (150 points)
  - Final exam (comprehensive): 25% (250 points)
- **Course Schedule**
  - The course schedule of topics and assignments will be posted on the course website.

### Homework, Tests, and Labs

- **Class participation and homework**
  - Students are expected to review topics taught in class, work on solutions to assigned problems, and be able to demonstrate skills and solutions during class. Homework assignments will be posted on Teams. Answers will be posted for selected problems.
  - To obtain full credit for the class participation and homework grade, you will need to attend 25 lecture sessions in-person (25 points), post questions about 25 different reading assignments (25 points), and present or post solutions to 5 homework problems (50 points). Questions about reading assignments must be posted before the class for which they are assigned and cannot be made up later.
  - Please use Teams for questions and discussion about course topics and assignments.

- **Exams and quizzes**
  - Quizzes
    - Quizzes assess students' abilities to demonstrate knowledge, design solutions to realistic problems, and present these solutions in a clear and professional fashion.
    - Quizzes can cover any concept or skill previously taught in the course, are generally offered at the rate of one per week.
  - Midterm and Final Exams
    - The midterm exam will occur approximately midway through the course.
    - The final exam date is scheduled by the university based on lecture time.
    - Like quizzes, exams may cover any concept or skill previously taught in the course.
  - The following are prohibited during quizzes and exams unless unambiguous and explicit permission is provided by the instructor:
    - Collaboration or communication with others.
    - Looking up answers in books, on the Internet, etc.
  - Blackboard will be used for quizzes and to keep track of grades.
- **Labs**
  - Labs are intended to provide an opportunity for students to explore and practice the use of tools and to apply concepts presented in class within the context of programming projects.
  - Lab assignments will be posted on Teams.
  - Lab assignments will be graded during in-person demonstrations with individual students.
    - Demonstrations must occur within one week following the lab due date.
    - Demonstrations may require students to modify their programs and demonstrate competency with development tools.
  - Students are expected to act professionally
    - By reading and studying relevant resources
    - 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 lab projects. There is no penalty for collaboration as long as it is documented.

### **Disabilities and Accommodations**

- If you have a disability and accommodations, please contact The Center for Accommodations and Support Services (CASS) at 747-5148, or by email to [cass@utep.edu](mailto:cass@utep.edu). For additional information, please visit the CASS website at [www.sa.utep.edu/cass](http://www.sa.utep.edu/cass).

### **Academic Honesty**

- Students are expected to conduct themselves in a professional and courteous manner, as prescribed by the Standards of Conduct: <https://www.utep.edu/hoop/section-2/student-conduct-and-discipline.html>

- Submitted 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.
- 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.
- If academic dishonesty is suspected: The Dean of Students office will be contacted for adjudication. A temporary "incomplete" grade will be issued if their investigation extends beyond the grading period.

### **Class Recordings**

Class recordings will enable you to have access to class lectures, group discussions, and review sessions in the event you miss a class meeting due to illness or other extenuating circumstance. Our use of such technology is governed by the Federal Educational Rights and Privacy Act (FERPA) and UTEP's acceptable-use policy. A recording of class sessions will be kept and stored by UTEP, in accordance with FERPA and UTEP policies. Your instructor will not share the recordings of your class activities outside of course participants, which include your fellow students, teaching assistants, or instructional assistants, and any guest faculty or community-based learning partners with whom we may engage during a class session. **You may not share recordings outside of this course.** Doing so may result in disciplinary action.

### **Guidance on Artificial Intelligence**

The use of generative AI tools such as Chat GPT is permitted in this course for the following activities, which must be noted or cited:

- Discovering algorithms for solving problems.

However, you may not use AI tools to complete the following activities:

- Generating code for programming assignments.

Students must cite any borrowed content sources to comply with all applicable citation guidelines, copyright law, and avoid plagiarism. Instances that violate these guidelines will be referred to the Office of Student Conduct and Conflict Resolution.

### **Plagiarism Detecting Software**

Some of your course work may be submitted to the SafeAssign or MOSS plagiarism detecting software. These tools are used to review submissions for originality and will help you learn how to properly attribute sources.

### **Copyright Statement for Course Materials**

All materials used in this course are protected by copyright law. The course materials are only for the use of students currently enrolled in this course and only for the purpose of this course. They may not be further disseminated.

## **Illness Precautions**

Please stay home if you have symptoms of a communicable illness. If you are feeling unwell, please let me know as soon as possible, so that we can work on appropriate accommodations.

## **UTEP Resources**

UTEP provides a variety of student services and support:

### Technology Resources

- [Help Desk](#): Students experiencing technological challenges (email, Blackboard, software, etc.) can submit a ticket to the UTEP Helpdesk for assistance. Contact the Helpdesk via phone, email, chat, website, or in person if on campus.

### Academic Resources

- [UTEP Library](#): Access a wide range of resources including online, full-text access to thousands of journals and eBooks plus reference service and librarian assistance for enrolled students.
- [University Writing Center \(UWC\)](#): Submit papers here for assistance with writing style and formatting, ask a tutor for help and explore other writing resources.
- [RefWorks](#): A bibliographic citation tool; check out the RefWorks tutorial and Fact Sheet and Quick-Start Guide.

### Individual Resources

- [Military Student Success Center](#): Assists personnel in any branch of service to reach their educational goals.

## **Course Schedule**

The course lecture schedule will be posted at <http://svmoore.pbworks.com/>

## Appendix. CS 3432 Learning Outcomes

### Level 1

- A1. Define and explain the purpose of an instruction set architecture (ISA).
- HSI2. Explain the relationship and differences between a high-level programming language, assembly language, and machine language.
- HSI3. Describe the fetch-execute cycle in terms of the hardware-software interface between machine instructions and processor components.
- NR1. Explain the relationship between a high-level language basic data type (e.g., signed or unsigned integer, floating point number) and its representation as a bit pattern inside the computer.
- A2. Describe the basic components of a processor (e.g., register file, special-purpose registers, control unit, memory) and how they interact with one another.
- HSI4. Explain how procedures are supported by processor hardware.
- HSI5. Explain exception/interrupt handling in terms of the hardware-software interface.
- HSI6. Explain various ways an operand can be addressed in an assembly language instruction.
- HSI7. Describe the process of compiling/assembling, linking, loading, and executing a program.

### Level 2

- NR2. Convert between different integer data representations (e.g., decimal, binary, hexadecimal, octal).
- NR3. Interpret the bit representation of a floating-point number.
- NR4. Perform addition and subtraction on two's complement representation of integers.
- NR5. Use bitwise operators to access and manipulate values stored in a subset of bits within a byte or word.
- NR6. Determine range and precision (if applicable) of numbers that can be stored for a given data type and determine whether an integer operation will result in overflow.
- HSI8. Convert between machine and assembly language representations of instructions – i.e., encode and decode instructions.
- HSI9. Trace the datapath through the processor for a given class of instructions (e.g., arithmetic-logical, memory access, conditional branch)
- HSI10. Trace the execution of an assembly language program with procedure calls in terms of allocation and deallocation of stack frames.
- L1. Translate expressions and assignment statements from C to assembly language.
- L2. Translate Boolean logic and control flow constructs (decisions, loops) from C to assembly language.
- L3. Translate operations on arrays, structs, and pointers from C to assembly language.

### Level 3

- HSI11. Implement/debug simple imperative programs in assembly/machine language.
- HSI12. Write or call a procedure with local variables, parameters, and return value in assembly language.
- HSI13. Implement a simple interrupt handler.
- T1. Compose, compile/assemble, execute, and debug simple programs in a command-line environment, using appropriate modularization and multiple files.