

Computer Networks - UTEP CS4316 and CS5313

instructor: [Dr. Eric Freudenthal](#)

URL: <https://sites.google.com/site/networkprotocolscourse/>

Please sign up (important):

- I'm transitioning to using "Microsoft Teams" for communications related class.
 - Previously I used google groups for this and will use it as a backup, so please join the [class google group](#) and subscribe to "all email"
- Join the class' [Microsoft Team](#) (follow link).
 - It works better if you install the related Office 365 Apps on the computers you use ([it's free for students and faculty](#)).

Text:

- Required: (free & web-viewable)
 - Peter Dordal's [Introduction to Computer Networks](#)
- Resources:
 - Python
 - Stavros Korokithakis' Quick Start [Tutorial - Learn Python in 10 minutes](#)
 - David Beazley's [Python Network Programming](#) and Gordon McMillan's [Socket Programming HOWTO](#)
 - The Python Software Foundation's spectacular [Informal Introduction to Python](#). This is actually Chapter 3 of [The Python Tutorial](#). I recommend starting here, reading forward through (at least) Chapter 5, then jumping around the tutorial's Table of Contents as needed.
 - Go: Jan Newmarch's [Network Programming with Go](#)
 - C: [Beej's Guide to Network Programming](#), or you can [buy it from the author for \\$20](#)
- Optional: Peterson & Davie's encyclopedic (and cheap if you purchase an older edition), *Computer Networks*.

Software:

- Virtual Machine with all of the required software for this course available [here: arch 1 vm](#)

Teaching Assistant:

- TBA

Important: Join the google group for this course: <https://groups.google.com/d/forum/2018-spr-nets>

This is a course on computer networks from a computer science perspective. While we examine the entire software/hardware stack, our focus will be on the principles that drive the design of operating system and application software that communicate via networks.

Course Outcomes

Students who attend this course will be able to understand and apply, analyze the following in new contexts

1. Separation of communication abstractions into logical layers that provide complementary properties.
2. Differences between the ISO and IP layered model
3. Significant properties of IP protocols and algorithms used to implement them
4. Physical, network, and symbolic address spaces and how they interact with routing.
5. Thread- and event-driven communication programming models
6. Abstractions provided by Berkeley sockets
7. Classes of security properties, threat models, and how they are commonly achieved
8. Familiarity with architecture of common wired and wireless transport (including mobile)

Students who attend this course will be able to define and describe the implications of:

1. Common modulation and coding techniques
2. Characteristics of common physical media (Ethernet, WiFi, BT, mobile)
3. Bandwidth, latency, reliability metrics
4. Error detection, correction,
5. Properties of cryptographic hashes and cyphers.

This is elaborated in painstaking detail in the [4316 Learning Outcome Rubric Matrix](#)

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
 - Frequency & Deadlines
 - Labs will be assigned during the lab course

- Due two weeks after being assigned unless otherwise indicated.
- Intention:
 - Labs are intended to provide an opportunity for students to practice and explore the application of concepts presented in class within programming assignments.
- 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 select and design problem-solving approaches, but they may not present solutions to each other for duplication or paraphrasing.
- 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.
 - **Originality:** Any code not provided by the instructor must be "original work" by the student. As in written "essay" assignments,
 - Each instance of a specific algorithm, code fragment, comment, or explanation composed by or with another person must be individually documented.
 - The majority of "original work" must not be composed of verbatim or paraphrased copies of code or comments composed by somebody else.
 - **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**

Grading

Course grades are typically computed as $\min(\text{labs}, \max(\text{average}(\text{final}, \text{midterm}), \text{final}))$. Scores may be collated by skill before aggregation.

Expectations of UG/Grad Students

Both graduate and undergraduate students will may attend this course. Graduate students are expected to demonstrate a higher level of technical competency, analytical maturity, and communication skills than undergraduates as demonstrated

by (1) class participation, (2) exams and (3) lab assignments. Some laboratory assignments will have advanced sections that only graduate students will be required to submit. Finally, the [official course outcomes](#) specify a range of mastery levels of the topics covered in this course that must be demonstrated by students in order to earn high marks. In order to be assigned similarly high marks, MS candidates are expected to demonstrate higher levels of mastery.

Can I skip a lab or midterm exam?

All labs must be completed on time. Exceptions are rare and must be approved and documented by the instructor.

Midterms are for students' benefit because they provide mid-term feedback regarding competencies.

In general, midterms cannot be "made up" - instead final grades will be computed using other exams.

Late policy

There is no policy for late assignments because they will not be accepted except under exceptional circumstances explicitly approved by the instructor.

Course grade computation

Team Policies

- Some labs may be assigned to teams. However there is always an individual evaluation for each activity.
 - · Teams could be self selected according to each assignment. Additional instructions will be given later.
 - · All members must contribute for each assignment and need to show their own part in the team's report.
 - · Is strongly recommended that the team members have the following roles that shall be rotated each time:
 - · The scheduler / tracker must distribute the tasks fairly among the members and will verify that each member submits their own part on time. If needed will contact members that are delayed
 - · The recorder will consolidate all results and will submit the final report
 - · The reviewer must check all contributions from the team.

The lab problems require the collaboration of the whole team to arrive to the proper solution. Therefore they might require several iterations to arrive to a proper conclusion.

Lab Policies

- You will be required to design and implement several lab assignments over the course of the semester.
 - Unless otherwise specified, lab assignments may be written in any language (though they're generally easiest in Python)
 - Lab reports will be committed to Github before each deadline.
 - The Lab grading criteria is approximately split equally among: function, design, and presentation (including documentation).
 - Additional requirements may be stated in the assignments.
 - Labs will only be accepted for grading up to one week after the posted deadline.

Academic Honesty

It is expected that the students will conduct with integrity in all course areas. Do not attempt to engage in a dishonest activity such as copying, plagiarism, falsifying information, etc. The professor will take measures to prevent such instances and will bring a case to the authorities.

Information about University wide policies could be found in the Dean of Students Web page at <http://studentaffairs.utep.edu/Default.aspx?alias=studentaffairs.utep.edu/dos>

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.