- index
  - Bioinformatics
  - ComputationalSolving
  - ComputerNetworking
    - Content
    - Syllabus
  - ContactPublicKey
  - DataStructures
  - DataStructuresLab
  - DeepLearning
  - FutureClasses
  - ProgrammingCpp
  - Security

Syllabus

Communication procedures (actually read and follow these)

- If you have a question regarding class administration or missing class, then please email the professor, or ask in person.
- If you have questions above-and-beyond the material, or just because you are interested, contact me however you want!
- If you have a question about theoretical course content, or assignment specifications, do NOT email, but instead, you should either:
  1. Come to the office hours of the professor, LEAD tutor, or grader(s). This option is guaranteed to at least get you a sincere attempt at an answer.
  2. Post your question in the discussion forum in Canvas, which is monitored by the graders and the professor (with no guarantees about answers being provided, though we very much try). I encourage students to help others in the forum, with the exception that it would not be wise to post or copy code or assignment answers in the discussion forum.
- If you have a question about how an assignment was graded, come to office hours. Neither the professor nor the graders can feasibly respond to grading issues via email (they almost always require real discussion and demonstration).
- If you have a debugging or related question about your programming assignments, then attend either the professor's, grader's, or LEAD tutor's office hours. It is not feasible to fairly and equally satisfy all requests for programming or debugging help via email. Specifically, do not send your code to the graders, LEAD tutors, or professor and expect that it is reasonable for them to debug it for you. Hint: this means you should plan to do your assignments early, to get debugging help if needed during office hours.
- If you truly can not make office hours times due to a consistent time conflict, I will make every attempt to find a time to meet with you ASAP. Please avoid abusing this policy merely because you do not feel like coming during

office hours times.

Professor

Dr. Taylor
http://cs.mst.edu/facultystaffandfacilities/facultydirectory/

Grading assistant

- Email: Ms. < at mst dot edu>

Course websites

- Grade recording and discussion forums: https://mst.instructure.com/
- All course materials: https://mst-cs.gitlab.io/
- Programming assignments: https://git-classes.mst.edu/

Help with the class material

There are a couple good sources of tutoring-style help in the class:

1. Instructor

Please feel free to come to my office hours, either during scheduled times or by appointment. If you are having trouble, this can be very helpful!

- Comp Sci room 212/213 (CS-Linux computer lab/lounge)
  - Times:
    - Starting the second week of class
    - Monday and Tuesday from 2-2:30 (dedicated hours which are less busy)
    - Monday and Tuesday from 2:30-4 (open hours, including other classes; may be more busy)
- Or by appointment (in CS 341)

2. TAs

Our student assistants are happy to help with programming or go over material in a small group or one-on-one setting. If you can't make any of these times, but would like to schedule some programming help, please feel free to email the TAs to set up a time.

Class/teaching evaluation and improvement

Please let us know what you like about the class and how it can be improved!

Course description

This course focuses on the Internet and the general principles of computer networking. It covers the TCP/IP model from the application layer to the link layer in a top-down approach. It also exposes students to multimedia networking, network security, wireless and mobile networks. This course includes performance modeling and analysis, development and implementation of complex communication protocols. Prerequisite: A "C" or better grade in Comp Sci 3800.

Prerequisite

- Grade of "C" or better in Comp Sci 2500

Textbook

Though I normally try very hard to not require a "paid" textbook, this one truly is the best in the field, and is a great book (the 7th edition is preferred, though 6th should be acceptable):

- https://www.pearson.com/us/higher-education/program/Kurose-Computer-Networking-A-Top-Down-Approach-7th-Edition/PGM1101673.html
- https://www.amazon.com/Computer-Networking-Top-Down-Approach-7th/dp/0133594149/ref=sr_1_1?ie=UTF8&qid=1480109067&sr=8-1&keywords=Kurose
- We will do assignments from this book.

Two supplementary textbooks:

- http://cnp3book.info.ucl.ac.be/ (free, 2 different versions; 1st ed explicitly mimics the primary book)
- http://intronetworks.cs.luc.edu/

Attendance

- Attendance will be taken directly the first week and indirectly after.
    - We will use stochastic sampling methods in class to take attendance
- Missing classes will greatly diminish your chances for getting a good grade in this class.
- If you miss more than 5 classes, we may drop you from the class.

Technical assignments and projects

- These will start the second or third week of the semester
- You should expect around 1 technical assignment every 1.5 weeks
- These will generally be due Tuesday night at 23:59, with some exceptions
- Published experimental studies in the fields of research in cognitive psychology and education have shown that frequent (rather than sparse)

recall, is both more effective for learning, retention, and synthesis, and also encourages frequent smaller bouts of studying, rather than cramming.

Submitting your assignments via Git

To prepare for submitting assignments

1. Log into https://git-classes.mst.edu with your S&T login
2. Watch the videos here: https://git-scm.com/videos
3. Read Appendix E - Submitting homework with Git, in the Data Structures Lab manual: ../../DataStructuresLab/Content/tools-for-computer-scientists.pdf
4. Some optional extras include the full set of materials listed under the Version Control lab day here: DataStructuresLab:Content

Submit using the repositories created for each assignment at: https://git-classes.mst.edu/

Execute once:

```
$ git clone https://url-for-your-repository
```

Execute as many times as you like from within the directory/repository you cloned to your hard drive (just an example):

```
$ git status
$ git add *.cpp *.h *.hpp *.txt *.py
$ git add SUBDIRECTORY/*
$ git commit -m "Informative description of the commit"
$ git push
```

Do not add:

Compiled or generated files like a.out, your executable files, etc.
Put the name of these files in a text file named .gitignore

If you see your changes reflected on the git-classes site, you have submitted successfully.

If you work from different computers and want to synchronize, or we make changes to your repository:

```
$ git pull
```

Git cheat sheet: https://gitlab.com/gitlab-com/marketing/raw/master/design/print/git-cheatsheet/print-pdf/git-cheatsheet.pdf

Quizzes / Daily questions

- Published experimental studies in the fields of research in cognitive psychology and education have shown that frequent (rather than sparse) recall, is both more effective for learning, retention, and synthesis, and also encourages frequent smaller bouts of studying, rather than cramming.
- Thus, we will have daily quizzes. These are administered using a clicker-like system. To avoid having to pay for a clicker, we use the free Plicker (paper clicker) system (https://get.plickers.com/).
- The goal of these is to incentivize four things:
    - Showing up to class! You get 1/2 points for a wrong answer.
    - Regular distributed studying and reading rather than batched cramming. The easy daily questions are on pre-class reading material not yet lectured.
    - Staying awake in class... Movement is life!
    - Reduced-stress assessment of your understanding of material (less painful grading)

Grading

You will be graded based on assignments, projects, homework, and other miscellaneous activities. We reserve the right to factor in points for attendance related performance, participation, or efforts demonstrated during office hours.

Assignment grading

Programs and scripts will be graded (on a scale from 0 to 100) primarily on their correctness. Complete and correct output for every test input case is necessary for your grade.

- A non-compiling, non-running, or crashing (Seg-fault, core dump, etc) program or script will receive a score of 10.

If a program runs, then points will be deducted for each incorrect test case output. Points may also be deducted for:

- Missing name
- Incorrectly formatted output. (Presentation Error)
- Memory Leaks
- Specific types of inefficiency
- File format issues

It is expected that all of your work runs correctly in the specified Linux environment we are working with in class, in the exact manner we specify in the assignment description. If you were contracted to write code for a job, and it ran on your computer, but not your employer's as they needed, your work would be considered a failure. In that light, you are also responsible for submitting all text and source files encoded UTF-8, Unix delimited.

After grading any given assignment, if the assignment appeared to be too difficult for the class, we may normalize to the top student's performance (the student with the highest point rank will get a 100% / A). This can, by definition, only help your grade.

Overall grading

We grade using the following procedure (percentages for each category may change slightly):

- 100 points for each technical assignment within two major categories: Wireshark labs and Programming assignments; see weighting on Canvas (it may change slightly).
- 150 points for the final programming project
- Any miscellaneous points

Your final grade = percent of possible points above
Your letter grade = standard S&T letter-percentile mapping:

A : [90.00 - 100] %
B : [80.00 - 90) %
C : [70.00 - 80) %
D : [60.00 - 70) %
F : < 60 %

Grades will not be rounded; for example, if you have a 79.9, that is a C.

View your grades

You can check your grades on Canvas: http://canvas.mst.edu/

Makeup, re-grades, and late work

- If you make a mistake in your code (small, large, or whatever) leading to a bad grade, then you can re-submit your programming assignment within 5 days of grades being returned, for a possible 50% of your points back, with a no-decrease rule: Max(old_grade, Mean(new_grade, old_grade)).
- If you miss this deadline, then re-submissions will not be accepted. This includes asking for re-submissions at the end of the semester because your grade is too low.
- You only get one re-submission per assignment.

- For each first-try submission, we will give you summary feedback in your git repositories, which will not be highly detailed. It is your job to determine what went wrong by re-reading the assignment specifications and improving your unit testing. During office hours, for pre-re-submission queries, we will NOT directly test your code for you, but will provide help. For post-re-submission queries, we will help you find bugs directly and do all the re-runs you like!
- Late submissions will not be accepted, though can be submitted as a re-submission, if within the re-submission time-window. It would be wise to account for something unexpected popping up last minute, so try to finish early.
- If you have an S&T-acceptable documented reason (i.e., illness, death in the family, etc) for missing **in-class** events, please see the professor to discuss potential re-scheduling or accommodation.

Academic honesty

You're here to learn and better yourself! Write all your work in your own words, and write your own code. Do not copy-paste (plagiarize) from any source. If you are not sure, err on the side of caution and do your work independently. Occasional infrequent help from a friend when your are really stuck may be reasonable, though if that "help" is frequent enough that your collaboration results in almost identical code, it was too much collaboration for an assignment intended to be independent work (which all are unless explicitly assigned as group work).

If you are found to be engaging in any form of academic dishonesty, the most severe penalties permitted by the university will be enacted. Incidences will typically result in grades of 0 for the respective course components, as well as notification of the student's advisor, the student's department chair, and the campus undergraduate studies office. Further academic sanctions may be imposed as well in accordance with university regulations (http://academicsupport.mst.edu/academicintegrity/). Those who allow others to copy their work are also committing plagiarism and will be subjected to the same procedures.

The Honor Code can be found at this link: http://stuco.mst.edu/honor-code/. Page 30 of the Student Academic Regulations handbook describes the student standard of conduct relative to the University of Missouri System's Collected Rules and

Regulations section 200.010, and offers descriptions of academic dishonesty including cheating, plagiarism or sabotage (http://registrar.mst.edu/academicregs/index.html).

We check your assignments against each other with software that is VERY good at detecting similarities and differences between any text files, including your source files. These methods are difficult, if not impossible to trick. Please do not try to copy-paste, share sources directly, or write all your code in a group or pair for individual assignments; you will not like the consequences!

Attempting to deceive attendance checking procedures is considered academic dishonesty for ALL parties involved. For example, do not submit someone else's pre-lab or lab assignment for them because they are not attending class.

Burns & McDonnell Student Success Center

The Student Success Center is a centralized location designed for students to visit and feel comfortable about utilizing the campus resources available. The Student Success Center was developed as a campus wide initiative to foster a sense of responsibility and self-directedness to all S&T students by providing peer mentors, caring staff, and approachable faculty and administrators who are student centered and supportive of student success. Visit the SSC at 198 Toomey Hall; 573-341-7596; success@mst.edu; web: http://studentsuccess.mst.edu/

Accessibility and Accommodations

If you have a documented disability and would like accommodations in this course, please facillitate providing documentation to the professor as early as possible in the semester. Disability Support Services staff will need to send a letter to the professor specifying the accommodation you will need. It is the university’s goal that learning experiences be as accessible as possible. If you anticipate or experience physical or academic barriers based on disability, please contact Student Disability Services at (573) 341-6655, sdsmst@mst.edu, visit http://dss.mst.edu/ for information, or go to mineraccess.mst.edu to initiate the accommodation process. Please be aware that any accessible tables and chairs in this room should remain available for students who find that standard classroom seating is not usable.

Title IX

Missouri University of Science and Technology is committed to the safety and well-being of all members of its community. US Federal Law Title IX states that no member of the university community shall, on the basis of sex, be excluded from participation in, or be denied benefits of, or be subjected to discrimination under any education program or activity. Furthermore, in accordance with Title IX guidelines from the US Office of Civil Rights, Missouri S&T requires that all faculty and staff members report, to the Missouri S&T Title IX Coordinator, any notice of sexual harassment, abuse, and/or violence (including personal relational abuse, relational/domestic violence, and stalking) disclosed through communication including but not limited to direct conversation, email, social media, classroom papers and homework exercises. Missouri S&T's Title IX Coordinator is interim chief diversity officer Neil Outar. Contact him (naoutar@mst.edu; (573) 341-6038; Temporary Facility A-1200 N. Pine Street) to report Title IX violations. To learn more about Title IX resources and reporting options (confidential and non-confidential) available to Missouri S&T students, staff, and faculty, please visit http://titleix.mst.edu.

Classroom Egress Maps

http://designconstruction.mst.edu/floorplan/

Backlinks: index:ComputerNetworking