

Syllabus

Communication procedures

- If you have a question regarding class administration or missing class, then please email the professor.
- If you have a question about a specific assignment, how an assignment was graded, or your grade tabulations, then please email the TA/grader.
- 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 or the TA. This option is guaranteed to at least get you an 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 debugging or related question about your programming assignments, then attend either the TA's or professor'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 TA 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.

Professor

Dr. Taylor

<http://cs.mst.edu/facultystaffandfacilities/facultydirectory/>

Graders/TAs

Programming assignment design, grading, and questions

- Mr. Kuhn <jck244 at mst dot edu>
- Mr. Studer <jmsxw4 at mst dot edu>
- Mr. Kovar <mjk8t2 at mst dot edu>
 - Monday's from 3-4:30 pm in Comp Sci room 212/213 (CS-Linux computer lab/lounge)

- Mr. Scharf <msxfd at mst dot edu>
 - Thursday's 3:30-4:30 in Comp Sci room 212/213 (CS-Linux computer lab/lounge)

Course website

The course website is hosted here, at:

<https://web.mst.edu/~taylorpat/>

Office hours and programming help

Here are some 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:
 - Wednesday 4-5 pm
 - Friday 3-4:30 pm
- Or by appointment (in CS 341)

2. LEAD hours

Also, there are LEAD learning center (<http://lead.mst.edu/schedule/>) hours.

- Comp Sci room 212/213 (CS-Linux computer lab/lounge)
 - Tuesday and Thursday 3:30-5 pm
- Cenn 105
 - Monday 5-7 pm
 - Monday March 26th will be canceled due to a scheduling conflict

The Learning Enhancement Across Disciplines Program (LEAD) sponsors free learning assistance in a wide range of courses for students who wish to increase their understanding, improve their skills, and validate their mastery of concepts and content in order to achieve their full potential. LEAD assistance starts no later than the third week of classes. Check out the online schedule at <http://lead.mst.edu/assist>, using zoom buttons to enlarge the view. Look to see what courses you are taking have collaborative LEAD learning centers (bottom half of schedule) and/or Individualized LEAD tutoring (top half of the schedule). For more information, contact the LEAD office at 341-7276 or email lead@mst.edu.

Our LEAD tutor, Mr. Dunker <jad8w4@mst.edu> , has generously offered to accept appointments for tutoring help throughout the week, so if you would like some extra assistance programming, please feel free to email him.

3. TA

Two of our graders (hours listed above), great students from a previous semester, will hold hours in the Lounge (Comp Sci room 212). They is available for help both with understanding the material, and also with the programming homework. If you can't make any of these times, but would like to schedule some programming help, please feel free to email any TA to set up a time.

Class/teaching evaluation and improvement

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

Course description

This course continues the development of structured programming concepts and their use in program development. Stacks, queues, linked list, arrays, trees, sorting, and searching will be taught together with their use in implementations of a number of algorithms. This course extends the study of structured programming by presenting the concepts and implementations of various abstract data types (ADTs) including lists, stacks, queues, trees, and graphs. We will discuss the trade-offs associated with implementing these ADTs with alternative data structures. You will also learn about more advanced C++ constructs.

Prerequisite

- Grade of "C" or better in Comp Sci 1570
- For a refresher on your C++
 - Remember the videos here: <http://classes.mst.edu/compsci1570/>
 - This C++ tutorial is pretty good, and well outlined: <http://www.cplusplus.com/doc/tutorial>

Course goals

"As gold which one cannot spend will make no person rich, so knowledge which one cannot apply will make no person wise." -- Samuel Johnson. As this quote suggests, the primary goal of this course is to provide you broad familiarity with methods commonly employed both in many real-world applications, and also those which are used for more advanced methods. The secondary goal of the class is to practice actually implementing these methods, both to assist understanding, and also to increase retention.

Textbook

Data Structures and Algorithm Analysis

Edition 3.2 (C++ Version)

Clifford A. Shaffer

Available FREE here: [../Content/DSA_Shaffer2013.pdf](http://Content/DSA_Shaffer2013.pdf)

Required readings and activities will be assigned from this book.

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.

Programming assignments

These will start the second week of the semester, and you should expect around 1 programming assignment every week and a half.

- These will generally be due Monday morning at 8:00, with some exceptions
- Published experimental studies in the fields of research in cognitive psychology and education have shown that frequent (rather than sparse) practice, is both more effective for learning, retention, and synthesis, and also encourages frequent smaller bouts of studying, rather than cramming.

Develop, test, and submit your assignments using the department Linux systems either in the lounge/lab, or remotely:

<http://itrss.mst.edu/linux-support/>

<http://itrss.mst.edu/linux-support/user-documentation/>

1. Remotely, ssh into the department's IT Linux systems:

Where NN = computer number

\$ ssh yourlogin@rcNNxcs213.managed.mst.edu

or if you want to run a graphical application:

\$ ssh -X yourlogin@rcNNxcs213.managed.mst.edu

2. Develop and test your program extensively

```
$ g++ *.cpp -o yourbinexe -std=c++11
```

```
$ ./yourbinexe
```

or

```
$ ./yourbinexe <"sample_input_if_exists.txt"
```

```
$ valgrind ./yourbinexe <"sample_input_if_exists.txt"
```

```
$ valgrind --leak-check=full ./yourbinexe <"sample_input_if_exists.txt"
```

3. 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:

```
$ git status
```

```
$ git add *.cpp *.h *.hpp *.txt *.py
```

```
$ git commit -m "Informative description of the commit"
```

```
$ git push
```

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>

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.

Please read, or re-read the department's C++ coding standards:

http://web.mst.edu/~cpp/cpp_coding_standard_v1_1.pdf

Assignment grading

Programs 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 a full score.

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

If a program compiles and 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):

0.9 * your percent of the possible absolute total points for programming assignments

0.1 * your percent of the possible absolute total points for the final project

Your total = Sum of above

Your letter grade = standard S&T letter-percentile mapping:

A : [90.00 - 100] %

B : [80.00 - 89.99] %

C : [70.00 - 79.99] %

D : [60.00 - 69.99] %

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 and late work

- No technical assignment re-submission is allowed, except on the first assignment under limited circumstances.
- Late submissions will not be accepted. 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 B&MSSC at 198 Toomey Hall; 573-341-7596; success@mst.edu; facebook: www.facebook.com/SandTssc; web: <http://studentsuccess.mst.edu/>

Accessibility and Accommodations

If you have a documented disability and would like accommodations in this course, please meet with 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 Disability Support Services at

(573) 341- 6655, dss@mst.edu, or 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 the classroom 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: [Courses:DataStructures](#)