CSCI 241 Data Structures
Spring 2025 Syllabus

Course Staff and Meeting Information

Jim Deverick, instructor
McGlothlin-Street 313

Yuchen Wang, instructor
McGlothlin-Steet 118

Mei Zhang, instructor
McGlothlin-Street 330

Chuntian Chi, TA
Chenghao Du, TA
Afia Farjana, TA
Haoyuan Tian, TA

section 01: MWF 1300—1350 Boswell 201
section 02: TR 1100—1220 Blow 333
section 03: TR 1230—1350 Blow 332
section 04: TR 1400—1520 Blow 332

office hours
M 1430—1600 Wang virtual by appt.
https://cwm.zoom.us/j/9063706629

W 1430—1600 Wang McGl 118

W 0900—1030 Deverick virtual by appt.
https://cwm.zoom.us/my/jwdeve

R 1530—1700 Deverick McGl 313

TR 1630—1800 Zhang virtual
https://cwm.zoom.us/j/95629993328

The best means of reaching us is through private Piazza posts. We ask that you use this medium primarily for communications. Email is a particularly unreliable means of reaching us.

Required Materials

- Goodrich, Tamassia, and Goldwasser. Data Structures & Algorithms in Python. Wiley, 2013. ISBN 978-1-118-29027-9. This textbook is available online at no cost to you through Swem Library. If you prefer a hard copy, any legal format is fine provided you have access all semester. We do not use any digital content from the textbook that requires paid access.
- Course materials will be distributed through the course website as detailed in class. A link to the course website will be available from Blackboard.
- Most items will be submitted on paper or electronically through Gradescope. Follow the instructions for submissions carefully, **especially regarding file names**, which must match exactly (including case). Submissions that do not conform to specifications may not receive credit. Some assignments may have submission links that take you outside of Gradescope or Blackboard.
- Assuming they don't require fees, we will use Piazza for class discussion and questions. Different students often pose similar questions and answering questions in a public forum benefits more people, including those who might not have thought to ask the question. You will receive an e-mail invitation to Piazza; please accept it. Piazza has changed their model to one that asks for

voluntary contributions. This is a function of Piazza and is not under the control of the instructors. You are not asked by the instructors nor are you required by this course to contribute to Piazza.

- Programming in this course will be done in Python3 using the Anaconda Python stack. We will provide instructions and assistance for setting up Anaconda for Windows, Mac OS X, or Linux on your personal machine. **You must not use Python 2.7, which is often the default on personal computers**.
- Programming in this course is done using a text editor and the command line. **You may not use any integrated development environment when writing code in this course.** We suggest VSCode, which is available for free. Many students choose to purchase Sublime. Some use Notepad++ on Windows. For command line environments, Mac users will employ Terminal; Windows users will use PowerShell. Both are installed by default. If the editor you use provide a "run" or "go" button, please don't use it.
- In the event that virtual sessions become necessary, we will require that you sign into Zoom (our domain is cwm) and request that you enable video during class. For privacy, some choose to use virtual backgrounds or hang a sheet or other cloth behind their seats, both of which are welcome. Please don't use animated avatars or video backgrounds during class. Signing in allows us to maintain a record of attendance and helps secure our meetings, and we can teach much more effectively if we can see you. Any virtual sessions will be recorded and posted to Blackboard for your review.

Exam Information

See the course schedule at the end of this document for exam section details. Switching exam sections requires approval of an Assistant Dean for Undergraduate Education.

Readings, Practice Problems, and Quizzes

Regular reading assignments from the textbook and academic papers will appear in Blackboard alongside lecture content. Reading assignments are very important, as the textbook often explains things from different perspectives. There is material in the assigned readings that will appear on quizzes and the exam, regardless of whether that material is covered in class.

We will issue practice problems and/or practice quizzes in class and may issue additional practice material for completion outside of class periodically to reinforce the material we cover in class and in the textbook. Homework is not typically collected or graded, though we may present it in electronic form that provides correctness feedback.

You are expected to complete all materials, regardless of point values. **There is a very strong correlation between students who rigorously engage with the practice materials and those who do well overall in the course.**

There will be several short paper quizzes. The initial versions will be completed in class. If you are not satisfied with your performance, you may retake quizzes (with different problems) in several evening sessions that will be available throughout the semester. Quizzes can only be completed during the in-class or retake sessions. We do not permit individual retakes or make-up sessions.

Programming Projects and Slip Days

There is a significant amount of programming in this course. In addition to implementing the abstract data types that we discuss, you will write small applications that employ those data types to see how they are used. Most programming projects will take between two and three weeks to complete and they increase in complexity as the course moves forward. The submission environment will provide automated functionality tests; you may submit as many times as you like until the deadline to improve your functional score. Other aspects of your programming projects, such as performance, readability, and adherence to specifications will be evaluated manually. You are required to use Anaconda Python 3.x for this course. Note that many Linux distributions and all versions of Mac OS X ship with Python 2.7. You must not use that version; it is not compatible with Python 3.x. **Always test your submission as your final act before submitting. Students have lost significant points because of last-minute edits.**

Each person will receive an initial allocation of two slip days to adjust project deadlines without penalty. Each applied slip day extends the deadline for a project by exactly 24 hours and cannot be divided into smaller extensions. Budget slip days carefully, as they become valuable at the end of the semester. Unused slip days have no value. If you submit a project late without using slip days, you will be penalized 25% of the project's total value per 24-hour period or part thereof beginning one minute past the posted deadline. Finally, you may combine slip days and late penalties if you wish. Late penalties only apply when you do not apply slip days.

No other student should see the code that you create for this class.

Some programming project submissions also include brief prose sections meant to illustrate both your theoretical and your practical understanding of the structures we create and apply. We will provide questions or other prompts to stimulate your responses, which will constitute a significant portion of your project scores. The Writing Resources Center, located on the first floor of Swem Library, is a free service provided to W&M students. Trained consultants offer individual assistance with writing, presentation, and other communication assignments across disciplines and at any stage, from generating ideas to polishing a final product. To make an appointment, visit the WRC webpage www.wm.edu/wrc.

Submission expectations are clearly stated at the end of each assignment. Generally, we only accept `.pdf` and `.py` files that are correctly named. **Any deviation from these specifications, including file names or formats, will void your submission and you will not receive credit.**

We allow you to submit an unlimited number of times, but we only download and evaluate your most recent submission. **Be certain that submissions are complete. We will not collate files from multiple submissions, so partial submissions cannot be evaluated.**

Attendance, Accessibility, and Honor

You are expected to be present and in good hygienic condition at every class meeting. We may take roll, including electronically. Please have your ID with you at all class meetings. Missing more than two meetings is cause for concern and may result in a grade penalty.

It is the policy of William & Mary to accommodate students with disabilities and qualifying diagnosed conditions in accordance with federal and state laws. Any student who feels s/he may need an accommodation based on the impact of a learning, psychiatric, physical, or chronic health diagnosis should contact us privately to discuss your specific needs. Students will also need to contact Student Accessibility Services staff at 757-221-2512 or at sas@wm.edu to determine if accommodations are warranted and to obtain an official letter of accommodation. For more information, please see www.wm.edu/sas. You have our word that such matters will be handled professionally and with sensitivity. We associate no stigma with any form of disability.

Please keep in mind that you are bound by all aspects of the College's Honor Code, to which you have pledged your support. Academic dishonesty does not serve your education well, and we will bring instances of it to the attention of the Honor Council. We mean this. If you cheat in this class, we will pursue disciplinary action against you, which may include suspension or permanent dismissal from the College.

Generally, we maintain an empty-hands policy for collaboration. You are free to discuss concepts with your colleagues *currently enrolled* in CSCI 241 at W&M, but you must not have any materials in front of you when you do. **No person other than you, the instructors, the TAs, or the lab instructors or consultants should ever see your work.** This means that relatives, students at other schools, alumni of the course (other than consultants), and online platforms such as CourseHero, Chegg, and others are prohibited. If you discuss something with permitted individuals using a whiteboard or paper or anything electronic, you must erase or destroy any collaboratively created results immediately when finished and without recording it in any other form, including photography. The granularity of collaborative discussions must be coarse enough that two people walking away from a conversation can reasonably be expected to produce different code. **If you see**

another student's work or allow your work to be seen by another student, then you are both liable for an Honor Code violation that we will prosecute.

We reserve the right to employ automated tools to check for plagiarism in both programming and written work.

Grading Policies

In most cases, quizzes are scored in Gradescope. If you believe we have assigned a score incorrectly, you may request a regrade from within Gradescope. Be honorable in your requests; do not ask for credit that you did not earn.

Projects are evaluated using automated functionality tests and manual human evaluation with written (electronic) feedback. We employ blind grading; the graders do not know whose work they are evaluating. To enable blind grading, it is important that you not include any personally identifying marks anywhere in your submission except where specified.

Your grade in this course will be computed as follows:

Quizzes	40%
Programming Projects	40%
Final Exam	20%

Letter grades are issued based on the following rubric:

A	At least 94%	C	At least 74% but less than 77%
A-	At least 90% but less than 94%	C-	At least 70% but less than 74%
B+	At least 87% but less than 90%	D+	At least 67% but less than 70%
B	At least 84% but less than 87%	D	At least 64% but less than 67%
B-	At least 80% but less than 84%	D-	At least 60% but less than 64%
C+	At least 77% but less than 80%	F	Less than 60%

Letter grades are available in Blackboard throughout the course and are updated as items are returned.

We do not apply any curves until the end of the semester. We reserve the right to consider class participation in borderline grade decisions. Any attendance penalties are applied after your grade is computed.

Mental and Physical Well-Being

William & Mary recognizes that students juggle different responsibilities and can face challenges that make learning difficult. There are many resources available at W&M to help students navigate emotional/psychological, physical/medical, material/accessibility concerns, including:

- The W&M Counseling Center at (757) 221-3620. Services are free and confidential.
- The W&M Health Center at (757) 221-4386.
- For additional support or resources & questions, contact the Dean of Students at 757-221-2510.
- For a list of other resources available to students, see:

Covered Topics

The purpose of this course is to introduce you to abstract data representation. You will learn how to represent data for various programming problems, and how to analyze the performance impacts of your data representation decisions. Though algorithms are not the primary focus of this course, we will introduce some well-known examples in order to illustrate the merits and performance of various data structures. The general schedule for the course follows.

Course Week	Content	Read Before Class	Practice Before Class	Progress	Submit projects before 2359 Quizzes in class
0	Introductions and Course Overview				
1	1.1 Python Variables and Arrays and Project 1 Overview	Ch. 1			
1	1.2 Circular Arrays	5.1, 5.4, 5.5	Ckpt. 1.1	P1 Insertion sort complete	
2	1.3 Multidimensional Arrays	5.6	Ckpt. 1.2	P1 Selection sort complete	
2	1.4 Performance Analysis	3.1, 3.2, 3.3	Ckpt. 1.3	P1 Writeup begun	
Feb. 9	**PROJECT DEADLINE**				**Project 1**
3	1.5 Linked Lists	7.1, 7.2, 7.3	Ckpt. 1.4		**Arrays Quiz**
3	1.6 Linked Lists				
4	Project 2 Iteration, Exceptions, and Testing Overview			P2 append_element, __str__ complete with testing	

4	2.1 Stacks	6.1, 7.1.1	Ckpt. 1.6	P2 insert_element_at, get_element_at, remove_element_at complete with testing	
5	2.3 Recursion	Ch. 4	Ckpt. 2.1	P2 __iter__, __next__ complete with testing; P2 Applications complete; all bugs fixed	**Linked Lists Quiz**
5	2.2 Queues and Deques	6.2, 6.3, 7.1.2, 7.2.2, 7.3.2	Ckpt. 2.3		
Mar. 2	**PROJECT DEADLINE**				**Project 2**
6	Project 3 Overview		Ckpt. 2.2	P3 Array_Deque including __grow complete with test cases.	**Stacks Quiz**
6	2.4 Queue Statistics			P3 Linked_List errors fixed, Linked_List_Deque complete with test cases.	
7	3.1 Binary Trees	Ch. 8	Ckpt. 2.4	P3 Stack, Queue complete with test cases.	**Recursion Quiz**
7	3.2 Binary Search Trees	Ch. 8, 11.1	Ckpt. 3.1		
8	Project 4 Overview		Ckpt. 3.2	P3 Applications complete.	
8	3.3 Heaps	9.1, 9.2, 9.3, 9.4			

Mar. 30	**PROJECT DEADLINE**				**Project 3**
9	3.4 Huffman Trees	Academic Paper	Ckpt. 3.3	P4 insert_element and in_order complete with in_order test cases	**Expression Trees and Binary Search Trees Quiz**
9	3.5 Balanced Binary Search Trees	11.3, Academic Paper	Ckpt. 3.4	P4 pre_order, post_order complete with test cases	
10	3.5 Balanced Binary Search Trees		Ckpt. 3.5	P4 remove_element, get_height complete with get_height test cases	
10	Catch up/Review			P4 remove_element test cases	
Apr. 13	**PROJECT DEADLINE**				**Project 4 Milestone**
11	Catch up/Review			P4 __balance single rotations complete with test cases	
11	4.1 Hash Tables	10.2		P4 __balance double rotations complete with test cases	
12	4.2 B+Trees	15.3	Ckpt. 4.1	P4 to_list, applications complete	**Balanced Binary Search Trees Quiz**
12	Catch up/Review		Ckpt. 4.2		
Apr. 27	**PROJECT DEADLINE**				**Project 4 Final**
13	Data Structures in the Wild				

13	Final Exam Review Session				
May 5	**Section 03 FINAL EXAM 0900—1200**				
May 5	**Section 01 FINAL EXAM 1400—1700**				
May 6	**Section 04 FINAL EXAM 1400—1700**				
May 12	**Section 02 FINAL EXAM 1400—1700**				