04-630: DATA STRUCTURES ALGORITHMS FOR ENGINEERS

Instructor: Dr. George Okeyo, Associate Teaching Professor, CMU Africa.

Email: gokeyo@andrew.cmu.edu

Location: F205 - IPE.

Course credit: 12 units

Prerequisite: None

TAs:

- ✓ Mohamed Gaye - mohamedg@andrew.cmu.edu
- ✓ Joseph Fadiji - jfadiji@andrew.cmu.edu.
- ✓ Lowami Uwimana - ulowami@andrew.cmu.edu
- ✓ Tracy Karanja - tkaranja@andrew.cmu.edu
- ✓ Oloruntobi Madamori - omadamor@andrew.cmu.edu

COURSE SYLLABUS

Course Timings

- Mon, Wed: 0800 to 0950 Hrs CAT (lectures).
- Tue, Thur: 1200 to 1330 Hrs CAT (labs)
- Fri: 1300 to 1400 Hrs CAT(recitation)

This course will be conducted in-person and attendance is ***required***.

Instructor, TA, and Office Hours

Office Hours: 1600-1700 Hrs, Tuesday & Thursday.

TA office hours will be available on Canvas. Recitations will be conducted by TAs to assist students in programming assignments. It is important that you attend these sessions. We will also run labs on Tuesday and Thursday.

If you send any emails regarding the course to the TA or instructor, please include the following in your subject line: "**DSA-Spring 2026**". For example, a student may write subject as: "**DSA-Spring 2026 : Unable to compile code**". All of us get a lot of emails. Having this prefix in the subject line will ensure we can search and read your emails on time.

Course Description

This course will provide engineers a background on algorithms, data-structures, and their practical uses. An engineer who may already know programming, will be able to learn the foundational aspects of how to design more efficient code that can optimize memory and CPU footprint. The goal is to equip engineers with skills needed when building software intensive systems.

The course begins by delving into a few algorithms and data structures that are routinely used by programmers. We then explore more efficient implementations of these constructs. Abstract data types are introduced, and the course provides an in-depth treatment of applications like searching, sorting, lists, stacks, and queues. The course will also cover trees, graphs, and algorithmic strategies. Performance analysis and tractability of algorithms, time/space complexity, automaton, and computability will also be discussed. Time permitting, the course will touch upon multi-threading, logging, testing, verification, and validation.

The course will be graded primarily on the basis of several programming assignments given during the semester, quizzes, and a final exam conducted during exam week. All programming assignments will be in C++, and the lectures would involve studying C++ code. Prior knowledge of coding in C or C++ would help the student greatly. However, it should still be possible to learn and use C++ even for those students without prior knowledge.

Any academic integrity violations will attract sanctions in line with the academic integrity policy.

Text Books

- Introduction to Algorithms. 3rd Edition. Authors: Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L. The MIT Press (2009)
- The Algorithm Design Manual, Third Edition, Steven S. Skiena (2020)
- Alfred V. Aho, Jeffrey D. Ullman, and John E. Hopcroft, Data Structures and Algorithms.
- David Harel and Yishai Feldman, Algorithmics: The Spirit of Computing, Third Edition

Digital copies of the books are available through the CMU library.

Learning Objectives

After completing this course, students should be able to:

- Recognize and analyze critical computational problems, generate alternative solutions to problems, and assess their relative merits.
- Identify and analyze factors that influence algorithmic performance and memory consumption.
- Design, implement, and document appropriate, effective, and efficient data structures & algorithms for a variety of real-world problems.

Course Philosophy

This course is not just about understanding the theory behind algorithms. Our goal is also: better programming, attention to detail, and good implementation. Please note that there are some key aspects in the DSA course:

- Students cannot make assumptions that are not given in an assignment specification document.
- A lot of students these days do not have attention to detail. It is important for students to realize that MINOR errors also can add up, and lead to SLOPPY results. Therefore, avoid grammar errors, submit the most optimized code, and provide citations and references. ALWAYS!
- Testing for corner cases is critical. Even if a sample file is provided, a student must look at the assignment document and come up with their own test cases where the code should work.
- Implementing a solution that works is not sufficient. Memory and timing is VERY important in this course. Each assignment will have a percentage of the credit assigned to these two aspects. You should always justify how memory and timing was optimized. You will have to reflect on this for every assignment.

Assessment

- There are 5-7 programming assignments each of which will be allotted between 40 to 100 points. One assignment will have limited TA support. One assignment will be a group assignment.
- There will be DSA-specific quizzes, each having 30-50 points. Details about the quizzes will be provided on Canvas. There will be up to five 5-10 minutes quizzes for knowledge checks.
- There will be up to five 5-10 minutes quizzes for knowledge checks.

- All programming assignments will be in C++. You should set up the programming environment within the first week. Instructions for setting up the environment may be provided by the TAs.

- Students are urged to go through the instructions given for each assignment carefully. Use the time in classes / recitations / labs / office hours to clarify doubts.

- There will be a final exam during exam week.

- Some homeworks may require writeups. All writeups must be of publication quality and in PDF format. Template format will be provided. Most homeworks will include a reflection on memory and timing optimization.

- Unless otherwise stated, all homeworks, code, write-ups must be submitted on canvas as per the deadlines.

- Programming assignments for the course will be visible to students on canvas from Week 2. Please note the last date for each submission and plan accordingly. If any assignment requires you to write an article, that PDF file will need to be submitted on canvas. Source code will be submitted on Gradescope. If there is any exception, we will announce this on canvas. The TAs will be opening up Gradescope for submission at least ONE week before the assignment deadline. If the Gradescope website for an assignment is available earlier, the TA will announce this on canvas.

Your grade will be determined based on *individual and group assignments, exams (final exam and quizzes)* and *participation.* The breakdown is shown in Table 1.

Table 1: Assessment Breakdown

Assignment	**Weight**
Programming Assignments(individual)	50%
Programming Assignment(group)	15%
Final Exam	20%
Quizzes	10%
Participation	5%

Further details about the assessments are provided below:

Programming Assignments(individual)

4-6 programming assignments will be issued accounting for 50% of the final grade. These programming assignments test your programming and problem-solving skills within the context of optimal data structures and algorithms. Each programming assignment will include a reflection in which each student will write about optimizations made regarding memory and timing.

Programming Assignments(group)

One group programming assignment will be issued accounting for 15% of the final grade. The assignment will foster presentation skills and collaboration and teamwork.

Quizzes

Two sets of quizzes will be provided. One quiz will be taken during class time and will last between 40-50 minutes. Other quizzes (up to 5 in number) lasting 5-10 minutes will also be completed for knowledge checks and retention. The quizzes will be graded out of 10%.

Final Exam

A final exam will be issued at the end of the semester during the final exam period. The exam will test your knowledge and skills on all topics covered throughout the semester. It accounts for 20% of the final grade.

Participation

Finally, your participation will be graded at 5%. This includes classroom involvement, completion of surveys, reading the syllabus, completing faculty course evaluations (FCE), contributions and participation on Piazza, class attendance, etc. Classroom involvement includes participation in class discussions and group activities and responding to or asking questions. Considering that you may not get a chance to participate in all classroom activities each week, participation will be tracked throughout the semester. Acceptable participation will be grades 80% and above and 0% otherwise.

COURSE SCHEDULE

There are two lectures each week. The broad outline of topics discussed each week is provided in Table 2. There are two timetabled labs and a recitation. The classes will be interactive. Students are expected to ask questions and seek guidance for homework assignments.

Table 2: Lecture and Assessment Schedule

Week	Date	Topic Description for the Week – 24+ classes	Assessment Schedule
Week 1	Monday	Introduction	Assessment dates will be
	Wednesday	Representing Algorithms: pseudocode, UML, finite state machines, flowcharts	

Week 2	Monday	Complexity Analysis	published on Canvas.
	Wednesday	Sorting and Searching Algorithms: In-place: Bubble, Selection, insertion	
Week 3	Monday	Sorting and Searching Algorithms: Not in place: Quick sort, merge sort	
	Wednesday	Pointers and Memory Management in C++	
Week 4	Monday	Heroes Day- no class	
	Wednesday	Abstract Data Types (ADT)	
Week 5	Monday	Containers, Dictionaries, and Lists I	
	Wednesday	Stacks and Queues	
Week 6	Monday	Trees: Binary ADT, Binary Search tree	
	Wednesday	Balanced Trees: Adelson-Velskii and Landis (AVL) tree, Red-Black tree	
Week 7	Monday	**Quiz 1**	
	Wednesday	Trees: Optimal code tree, Huffman tree	
Week 8	Monday	Heaps: Priority queue, Binary heap, heapsort	
	Wednesday	Hashing: Dictionaries, Hashing, Hash functions, Collision resolution, Complexity, Applications	
Week 9	Monday	Graphs: Graph representation adjacency matrix and adjacency list	
	Wednesday	Graphs: BFS, DFS, topological sorting	
Week 10	Monday	Graphs: Min-Spanning tree: Prims/Kruskals,	
	Wednesday	Graphs: Shortest Path algos: Dijkstras, Floyds	
Week 11	Monday	Complex Networks	
	Wednesday	Complex Networks	
Week 12	Monday	Algorithmic strategies: Brute force, Divide and Conquer, Greedy algorithms	
	Wednesday	Algorithmic strategies: Dynamic programming, Combinatorial search and backtracking, Branch and bound	
Week 13	Monday	**No classes - 1994 Genocide Against the Tutsi Memorial**	
	Wednesday	**No classes - 1994 Genocide Against the Tutsi Memorial**	
Week 14	Monday	**Quiz 2**	
	Wednesday	B-Tree and Fibonacci heaps; Genetic algorithms: Finding solutions to polynomials, optimal path algorithms	
Week 15	Monday	Algorithm correctness, STL	
	Wednesday	Matrix operations: LU Decomposition, Sparse matrix multiplication, Matrix inversion and Least Squares computation	
Week 16	Monday	Final Examinations	
	Wednesday	Final Examinations	

GRADING SYSTEM FOR THE COURSE

Table 3: Grading Criteria

Percentage	Grade	Narrative description
92-100	A	indicates **exceptional work**
90-91	A-	indicates **excellent work**
87-89	B+	indicates **good, above average work**
82-86	B	indicates **good, competent work**
80-81	B-	indicates **above average work**
77-79	C+	indicates **slightly average but passing work**
73-76	C	Indicates **average but passing work**
70-72	C-	Indicates **below average and failing work**
60-69	D	indicates **unsatisfactory work**
Less than 60	R	indicates failing grade

NOTES ON COURSE POLICIES

1) Note on Slack Policy/Flex Days

Each student is automatically granted three (3) "Flex Days" to use throughout the semester on any assignment, for any reason, no questions asked. Conceivably, you could have one(1) flex day for each assignment, for up to three (3) assignments. To use a flex day, simply submit your assignment late. You do not need to email the instructor or TAs for permission. It is your responsibility to keep track of your used flex days.

Once all your flex days are exhausted, late submissions will incur point deductions as below:

- ✓ Up to 24 hours late: 10% deduction from the total possible points.
- ✓ Between 24 and 48 hours late: 20% deduction from the total possible points.
- ✓ More than 48 hours late (after all flex days are used): A grade of zero (0) will be assigned, and the work will not be accepted.

For significant, documented circumstances (e.g., hospitalization, family emergency, extended illness) that go beyond the flexibility provided by the "Flex Days," please contact the instructor via email as soon as possible to discuss potential accommodation. These will be evaluated on a case-by-case basis and may require documentation **(which you should only provide to the instructor upon request). Don’t send unsolicited documentation!**

2) A Note on Student Well-Being

We are all under a lot of stress and uncertainty during the semester. Make sure to move regularly, eat well, and reach out to your support system or me gokeyo@andrew.cmu.edu if you need to. We can all benefit from support in times of stress, and this semester is no exception.

We encourage you to seek a *healthy balance.* Universities are in general vibrant communities, places of tremendous vitality and richness that offer abundant opportunities for meaningful work and play. This abundance brings with it the challenge of maintaining a healthy, balanced life – a life characterized by productive tension among such competing needs as work and play, sleep and wakefulness, solitude, and sociability. All members of university communities – ***students***, staff, and faculty – have the responsibility to promote balance in their lives by making thoughtful and balanced choices.

Balanced choices flow from an understanding that human flourishing requires the fulfillment of very ***real physical, emotional, spiritual, and social needs***. Therefore, an understanding that failure is part of the road to success in all endeavors, whether academic, extracurricular, or social is critical to striking a balance between expectations of success and mitigating incidents of failure.

3) Note on Community Engagement

We commit to creating a safer, more inclusive environment that encourages students from diverse backgrounds and perspectives to realize their full potential during this course. This environment should inspire classroom discussion, and respectful and empathetic discourse where each participant is treated with respect and dignity.

4) Note on Accommodations for Students with Disabilities

If you have a disability and have an accommodations letter from the Disability Resources office, I encourage you to discuss your accommodations and needs with me as early in the semester as possible. I will work with you to ensure that accommodations are provided as appropriate. If you suspect that you may have a disability and would benefit from accommodations but are not yet registered with the Office of Disability Resources, I encourage you to contact them at access@andrew.cmu.edu.

5) Note on Academic Integrity

Students are encouraged to talk to each other, to the T.A.s, to the instructor, or to anyone else about any of the homework assignments. Any assistance, though, must be limited to discussion of the problem and sketching general approaches to a solution. Each student must write out his or her own homework solutions. Consulting another student's solution is prohibited, and submitted solutions may not be copied from any source. Copying from someone else's homework (present or past years or past year's solutions), lab write-up, or exam or allowing another student to copy his/her work, will be considered as cheating. Any form of collaboration is strictly prohibited on the exams and is considered cheating. If you have any question about whether some activity would constitute cheating, please feel free to ask**.**

If you *even suspect* that you have collaborated with any other person or taken help from online forums, or used material from elsewhere, list their name(s)/sources at the top of the first page of your assignment. Also send an email to the TA/instructor saying that you have used external resources for a particular assignment. **This email has to be sent BEFORE the assignment submission deadline. Sending this email does not excuse you from charges of plagiarism.** It may merely reduce the impact. Instead of getting failed in the course, you may be given a zero on the assignment.

Refer to the following CMU's Academic Integrity Policy at **Academic Integrity - University Policies** for further information. You are encouraged to review the three policy violations i.e., *cheating, plagiarism, and unauthorized assistance(including reference to electronic resources which for guidance in this course includes generative AI).*

You may not have much experience with CMU's Academic Integrity Policy and the issues are sometimes subtle. If you have any question about the right thing to do, **just ask**. Any of the faculty or teaching assistants can provide guidance. If you are uncomfortable asking, then you probably already know that you are violating the academic integrity policy and should change your actions accordingly.