CS 2414 – Data Structures – Spring 2026

Syllabus

Instructor: Dr. Sridhar Radhakrishnan, sridhar@ou.edu

Office Location: Sarkeys Energy Center, SEC 1178
Office Hours: 4:30 PM to 6:00 PM (Tuesday and Thursday).
Zoom Link:
https://oklahoma.zoom.us/j/95606704172?pwd=G7IFEG4VRaKvq2AVDZUQP18YdLoGpQ.1&from=addon

Class Location: SEC A235
Class Time: 6:00 PM to 7:20 PM (Tuesday and Thursday)

Course Pre: CS 2334 and (CS 2813 or MATH 2513) and (Math 1823 or MATH 1914)

Textbook: Radhakrishnan, S., Wise L., and Sekharan, N. 2013. *Data Structures Featuring C++: A Programmer's Perspective (Links to an external site.)* This textbook is available on Amazon.

Learning Management System/Website: https://canvas.ou.edu/courses/381982

Course Catalog Description: Representation, analysis and implementation of data structures and associated algorithms including: algorithm complexity, sorting algorithms, lists, stacks, queues, search trees (AVL, Red-Black, Splay, 2-3), Heaps, Graphs, and Hashing. Written communications required in some projects. Ethical issues and tools and techniques used in writing secure applications will also be discussed. The primary programming language is C++ with a debugging tool.

Course Format: This course will be taught in a traditional in-class format with both lecture and laboratory components. Lectures will focus on theory, algorithms, and applications of data structures, while labs will emphasize implementation and practice. Labs are scheduled on Mondays and will include both quizzes and hands-on lab projects. From time to time, video recordings of lectures may be made available on Canvas for supplemental review.

Course Activities and Requirements

- Quizzes: There will be approximately eight quizzes, administered during lab sessions. Quizzes are short assessments to reinforce recently covered material.
- Programming Projects: Students will complete four large programming projects, each assigned with an ~18-day completion period. These projects must be written in C or C++, with specifications provided for each assignment. Projects are to be completed individually.
- Lab Projects: Lab sessions will include programming tasks that must be completed and submitted before the end of the lab session. These are designed to strengthen practical understanding of data structures.
- Exams: Students will complete two midterm exams during the semester, each covering a significant portion of the material.
- Final Exam: The final exam will be comprehensive and will take place on December 11, from 6:00 PM to 8:00 PM. The final is mandatory; failure to take it will result in an automatic F for the course.
- Attendance: Attendance will be taken for both lecture sessions and lab sessions. Regular attendance and participation are expected.

Component	Percentage
Attendance — Lecture	6%
Quizzes and Lab Projects	24%
Exam 1	15%
Exam 2	15%
Final Exam	20%
Programming Projects (4 total)	20%

ABET Outcomes of Instruction in CS 2414:

1. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
2. Design, implement and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

Belonging Statement: The University of Oklahoma fosters an inclusive culture of respect and civility, belonging, and access, which are essential to our collective pursuit of excellence and our determination to change lives. The unique talents, perspectives, and experiences of our community enrich the learning, and working environment at OU, inspiring us to harness our innovation, creativity, and collaboration for the advancement of people everywhere.

CS 2414 is a learning community where curiosity and interaction are highly valued. As your instructor, I strongly believe that asking questions and engaging in discussions are essential for mastering Data Structures. Students are encouraged to actively participate, challenge concepts, and collaborate, as this subject is best learned through dialogue and shared exploration.

Course Expectations: Students are expected to actively engage in the learning process by attending lectures and labs, participating in discussions, and seeking clarification on challenging concepts. Data structures form a core foundation of computer science, and mastery requires both theoretical understanding and hands-on implementation.

1. Programming and Assignments
   - Programming projects are central to learning in this course. There will be four major programming projects, each assigned with about 18 days to complete. Projects are challenging and require steady progress — they cannot be completed at the last minute.
   - All programming projects must be completed independently in C or C++, adhering to the specifications provided.
   - Submissions must follow proper coding standards, include clear documentation, and demonstrate correctness through adequate testing.
   - Projects must be submitted on time via Canvas. Late submission policies will follow the rules in the syllabus.
   - Students may use LLMs and AI tools (e.g., GitHub Copilot, ChatGPT) for assistance, but they remain fully responsible for correctness. All AI use must be acknowledged, and prompts must be submitted along with the project.
2. Exams and Quizzes
   - The course includes two midterm exams, about eight quizzes (given during labs), and a comprehensive final exam.

- All exams and quizzes will be in-class, on paper, and without electronic devices.
- The final exam is mandatory; failure to take it will result in an automatic F for the course.
- Success in exams and quizzes requires keeping up with lecture material, lab practice, and problem-solving exercises.

3. Collaboration and Academic Integrity
   - Students may discuss concepts and problem-solving strategies with peers, but all programming work must be written independently.
   - Sharing code, copying from the internet, or submitting work generated by others constitutes academic misconduct and will be reported.
   - Any external resources or peer discussions must be documented in project submissions.
4. Time Management and Work Ethic
   - Students should begin programming projects early, allocate sufficient time for debugging, and avoid last-minute work.
   - Consistent review of material, practice in labs, and proactive help-seeking will contribute significantly to success.
   - Some project specifications may not contain every detail; students are expected to clarify requirements during lectures, labs, or office hours.
5. Classroom and Lab Conduct
   - Laptops and devices should only be used for course-related activities during lectures and labs.
   - Respectful and professional conduct is expected at all times.
   - Attendance will be taken for both lectures and labs. Lab attendance counts toward the lab grade, and lecture attendance counts separately.
   - Active participation in both lectures and labs will enrich the learning experience.
6. Seeking Help
   - Office hours are available for one-on-one guidance.
   - TAs and graders can provide help with concepts, logic, and debugging strategies, but they are not responsible for fixing student code.
7. Critical Thinking and Problem-Solving
   - Students will analyze and implement data structures such as linked lists, stacks, queues, trees, graphs, and hashing.
   - Problem-solving skills are essential for applying data structures to real-world computing challenges.
   - Ethical considerations and secure programming practices will be emphasized throughout.

Course Goals: The goals of this course are to:

1. Provide a Strong Theoretical Foundation in Data Structures
   - Introduce students to the fundamental principles, design, and analysis of data structures.
   - Cover key topics such as arrays, linked lists, stacks, queues, trees, graphs, hashing, and sorting algorithms.
   - Develop an understanding of algorithm complexity and its impact on data structure performance.
2. Develop Practical Programming Skills for Implementing Data Structures
   - Enable students to write, implement, and test various data structures in C or C++.

    - Reinforce theoretical concepts through hands-on programming projects that require independent problem-solving.
    - Emphasize debugging techniques, code documentation, and adherence to programming best practices.
3. Enhance Problem-Solving and Algorithmic Thinking
    - Teach students how to select and apply appropriate data structures for different computational problems.
    - Improve the ability to analyze trade-offs in efficiency, memory usage, and time complexity.
    - Foster an understanding of how data structures impact software performance and scalability.
4. Build Competency in Advanced Data Structures and Their Applications
    - Introduce students to self-balancing trees (AVL, Red-Black, Splay, 2-3 trees), heaps, and priority queues.
    - Cover graph representations, traversal algorithms, and shortest path algorithms.
    - Explore hashing techniques and their applications in database indexing and security.
5. Expose Students to Secure and Ethical Programming Practices
    - Introduce best practices for secure coding and data protection when implementing data structures.
    - Discuss ethical considerations in software development, including responsible AI use in coding.
6. Develop Critical Thinking for Data Structure Design and Optimization
    - Encourage students to evaluate trade-offs in data structure selection based on efficiency and application constraints.
    - Foster an analytical mindset for optimizing algorithms and minimizing memory usage.
7. Prepare Students for Advanced Courses and Industry Applications
    - Equip students with the foundational knowledge required for advanced coursework in algorithms, databases, AI, and software engineering.
    - Provide skills relevant to real-world software development, including performance optimization, debugging, and problem-solving.

Course Learning Objectives: Upon successful completion of this course, students will be able to:

1. Understand Core Data Structure Concepts
    - Explain the fundamental principles, design, and use cases of essential data structures, including arrays, linked lists, stacks, queues, trees, graphs, and hash tables.
    - Analyze the time and space complexity of data structures using Big-O notation.
2. Implement Data Structures in C/C++
    - Develop efficient implementations of various data structures in C or C++, using dynamic memory allocation, pointers, and object-oriented programming principles.
    - Apply proper debugging, documentation, and coding standards when developing projects.
3. Design and Analyze Algorithms for Data Structures
    - Implement and analyze searching and sorting algorithms, including binary search, quicksort, mergesort, and heapsort.
    - Compare the trade-offs between different data structures and algorithms to solve computational problems effectively.
4. Apply Linked Lists, Stacks, and Queues

- Implement and manipulate singly linked lists, doubly linked lists, and circular linked lists.
- Develop stack and queue implementations and apply them to real-world computing problems, such as expression evaluation and scheduling.

5. Work with Trees and Self-Balancing Trees
    - Implement binary search trees (BST), AVL trees, Red-Black trees, Splay trees, and 2-3 trees.
    - Understand tree operations, including insertion, deletion, traversal (inorder, preorder, postorder), and their impact on performance.
6. Understand and Implement Hashing Techniques
    - Explain and implement hash tables, collision resolution techniques (chaining, open addressing), and hash functions.
    - Apply hashing in real-world applications such as database indexing and caching.
7. Analyze and Apply Graph Data Structures
    - Implement graph representations (adjacency list, adjacency matrix) and apply graph traversal algorithms like BFS (Breadth-First Search) and DFS (Depth-First Search).
    - Apply graph algorithms for solving shortest path (Dijkstra's, Floyd-Warshall), minimum spanning tree (Prim's, Kruskal's), and network flow problems.
8. Evaluate and Optimize Data Structures for Performance
    - Analyze the trade-offs between different data structures in terms of speed, memory usage, and computational efficiency.
    - Optimize data structures for practical applications, considering best-case, worst-case, and average-case complexity.
9. Apply Data Structures to Real-World Problems
    - Solve real-world computing challenges using appropriate data structures, including searching, sorting, memory management, and scheduling.
    - Recognize and implement secure and ethical programming practices in data structure design.
10. Develop Critical Thinking and Problem-Solving Skills
    - Identify and apply appropriate data structures for various computational problems.
    - Improve problem-solving skills through programming assignments, debugging, and project-based learning.

Course Reflection Survey: You'll receive a Course Reflection Survey at the end of each semester for each course that you are enrolled in. I strongly encourage you to complete this survey. Your feedback can help me adjust my class for future semesters to help other students be successful. Your feedback is confidential and I will only receive it after final grades are due. Course Reflection Survey results may also factor into teaching evaluations and annual performance reviews and are shared with department and program chairs.

Copyright Statement: Sessions of this course may be recorded or live-streamed. These recordings are the intellectual property of the individual faculty member and may not be shared or reproduced without the explicit, written consent of the faculty member. In addition, privacy rights of others such as students, guest lecturers, and providers of copyrighted material displayed in the recording may be of concern. Students may not share any course recordings with individuals not enrolled in the class or upload them to any other online environment.

Tentative Course Schedule—Note that some lectures will be recorded video lectures, and we will announce this in advance.

Week	Date	Activity	Notes
1	Mon, Jan 19, 2026	Martin Luther King Holiday	—
	Tue, Jan 20, 2026	Lecture: C++ Programming	Project 1 Assigned
	Thu, Jan 22, 2026	Lecture: C++ Programming	—
2	Mon, Jan 26, 2026	Lab: Orientation, C++ environment setup	—
	Tue, Jan 27, 2026	Lecture: Chapter 1 — OOP Intro	—
	Thu, Jan 29, 2026	Lecture: Chapter 1 — OOP Intro	—
3	Mon, Feb 2, 2026	Lab: OOP practice (class syntax, ctors/dtors, const methods)	Quiz 1 (C++/OOP)
	Tue, Feb 3, 2026	Lecture: Chapter 2 — Algorithms & Recursion	—
	Thu, Feb 5, 2026	Lecture: Chapter 2 — Algorithms & Recursion	—
4	Mon, Feb 9, 2026	Lab: Recursion practice	Quiz 2 (Recursion)
	Tue, Feb 10, 2026	Lecture: Chapter 3 — Arrays, Strings, Vectors	—
	Thu, Feb 12, 2026	Lecture: Chapter 3 — Arrays, Strings, Vectors	Project 1 Due; Project 2 Assigned
5	Mon, Feb 16, 2026	Lab: Arrays/Strings/Vectors practice	Quiz 3 (Arrays/Strings/Vectors)
	Tue, Feb 17, 2026	Lecture: Chapter 4 — Linked Lists	—
	Thu, Feb 19, 2026	Lecture: Chapter 4 — Linked Lists	—
6	Mon, Feb 23, 2026	Exam 1 Review Lab (covers OOP → Arrays/Strings/Vectors/Linked Lists)	—
	Tue, Feb 24, 2026	**Exam 1 (covers OOP → Arrays/Strings/Vectors/Linked Lists)**	—
	Thu, Feb 26, 2026	Lecture: Chapter 5 — Stacks & Queues	—
7	Mon, Mar 2, 2026	Lab: Stacks & Queues practice	Quiz 4 (Linked Lists); Project 2 Support
	Tue, Mar 3, 2026	Lecture: Chapter 5 — Stacks & Queues	Project 2 Due; Project 3 Assigned
	Thu, Mar 5, 2026	Lecture: Chapter 6 — Binary Trees	—
8	Mon, Mar 9, 2026	Lab: Stack/Queue review & Binary Trees intro	Quiz 5 (Stacks/Queues)
	Tue, Mar 10, 2026	Lecture: Chapter 6 — Binary Trees	—
	Thu, Mar 12, 2026	Lecture: Chapter 6 — Binary Trees	—
9	Mon, Mar 16, 2026	Spring Break	
	Tue, Mar 17, 2026	Spring Break	
	Thu, Mar 19, 2026	Spring Break	
10	Mon, Mar 23, 2026	Lab: Binary Tree practice (insert, traversals, height)	Quiz 6 (Binary Trees)
	Tue, Mar 24, 2026	Lecture: Chapter 7 — Self-Balancing Search Trees	—

	Thu, Mar 26, 2026	Lecture: Chapter 7 — Self-Balancing Search Trees	Project 3 Due; Project 4 Assigned
11	Mon, Mar 30, 2026	Lab: AVL rotations, balance factors	Quiz 7 (Self-Balancing Trees); Project 3 Support
	Tue, Mar 31, 2026	Lecture: Chapter 8 — Priority Queues/Heaps	—
	Thu, Apr 2, 2026	Lecture: Chapter 8 — Priority Queues/Heaps	—
12	Mon, Apr 6, 2026	**Exam 2 Review Lab (covers Stacks, Queues, Binary Trees, Self-Balancing Trees, Heaps)**	—
	Tue, Apr 7, 2026	Exam 2 (covers Stacks, Queues, Binary Trees, Self-Balancing Trees, Heaps)	—
	Thu, Apr 9, 2026	Lecture: Chapter 9 — Sorting	—
13	Mon, Apr 13, 2026	Lab: Sorting practice (merge/quick)	Quiz 8 (Sorting)
	Tue, Apr 14, 2026	Lecture: Chapter 9 — Sorting	—
	Thu, Apr 16, 2026	Lecture: Chapter 10 — Hashing	—
14	Mon, Apr 20, 2026	Lab: Hashing practice (functions, collisions)	Project 4 Support
	Tue, Apr 21, 2026	Lecture: Chapter 10 — Hashing	Project 4 Due
	Thu, Apr 23, 2026	Lecture: Chapter 11 — Graphs	—
15	Mon, Apr 27, 2026	Lab: Graphs practice (adjacency list, BFS/DFS)	Quiz 9 (Graphs — Basics)
	Tue, Apr 28, 2026	Lecture: Chapter 11 — Graphs	—
	Thu, Apr 30, 2026	Thanksgiving — No Class	—
16	Mon, May 4, 2026	Lab: Graphs practice (shortest paths, connected components)	Optional check-in
	Tue, May 5, 2026	Lecture: Review for Final Exam	—
	Thu, May 7, 2027	**Final Exam (6:00-8:00 PM, SEC A235)**	—

Note that May 3-10, 2025, is a day during the dead week. Since our course is an evening course, the final exam is to be administered during the last day of the class which is May 7, 2026.

Course Policies

Programming Projects:

Projects must be coded in C or C++. We will use ANSI C or C++, so if your program compiles with any G++ or GCC compiler, you are set to go.

1. You will also use the Gradescope facility to submit the source program.
2. For every 24 hours late, you will be deducted 10% of the grade of the programming project. Any project that is more than 5 days late will not be evaluated.
3. A programming project that does not meet the specifications will receive an automatic 50% grade deduction.
4. You are better off submitting a working project on the fifth day rather than one that does not work on the due date.
5. Programs have to be documented clearly. Programs that lack or are weak in documentation will receive a deduction of up to 30% of the grade. Follow the documentation methods used in programs in your data structures book.
6. You will demo your project to the grader during the grader-assigned special office hours **if the grader so wishes**. Graders are not responsible for debugging your programs.
7. The project specifications presented by the instructor may not contain all the implementation details. It is your responsibility to understand the specifications thoroughly. Please ensure that all relevant questions regarding the project are asked during class time.
8. **Copying programs or consulting others for coding is strictly prohibited and will be treated as plagiarism. Additionally, copying programs from the Internet is also strictly prohibited. All projects are individual projects; hence, you are required to work independently without help from others.**
9. In addition to the above general evaluation policies, each programming project will have a set of specifications that must be met.

Use of LLM and AI Assistance: Students are permitted to use Generative AI tools, such as ChatGPT and GitHub Copilot, to assist in understanding data structure concepts, debugging code, and generating ideas for programming assignments. However, all submitted work must reflect the student's own understanding and effort. While AI can be a valuable tool, it should complement learning rather than replace critical thinking and hands-on programming experience. If AI is used to assist in coding, problem-solving, or explanations, students must explicitly acknowledge its use by adding comments in their code or submissions detailing what aspects were AI-assisted. Additionally, as part of the project evaluation, students must submit any prompts they used to generate or refine their code as a separate document. Since AI-generated code may not always be correct or optimal, students are responsible for ensuring their final submission is functional, accurate, and meets all project requirements. Directly copying AI-generated solutions without comprehension or modification is not acceptable and may be considered academic misconduct. Students should consult the instructor if they have any questions about appropriate AI usage.

Attendance Policy: Regular attendance is essential for success in this course, as active participation in lectures and discussions is crucial to understanding operating system concepts. Attendance will be recorded in each class using Top Hat, and it will account for 15% of the overall course grade. Students are responsible for ensuring their attendance is accurately recorded. Absences due to emergencies or university-approved events must be communicated in advance when possible, and

proper documentation may be required for consideration. Missing class without a valid excuse may negatively impact the final grade. While lecture recordings may be available for some sessions, they are not a substitute for in-class participation. Students are encouraged to engage actively in discussions and ask questions, as interaction is a key component of learning in this course.

Exams: All exams will be conducted on paper in class. All exams will be closed books, notes, laptops, or any electronic devices. Failure to take the final exam will result in an automatic F as the course grade.

Incompletes: The grade of "I" is intended for the rare circumstance when a student who has been successful in a class has an unexpected event shortly before the class's end. We will not consider giving a student a grade of "I" unless the following three conditions have been met:

1. It is within two weeks of the end of the semester.
2. The student has a grade of C or better in the class.
3. The reason that the student cannot complete the class is properly documented and compelling.

Classroom Conduct: Because cell phones and laptops can distract substantially from the classroom experience, students are asked not to use either during class except when required as part of a classroom exercise. Disruptions of the class will also not be permitted. In the case of disruptive behavior, we may ask that you leave the classroom and charge you with violating the Student Code of Responsibilities and Conduct. Examples of disruptive behavior include:

- Allowing a cell phone or pager to repeatedly beep audibly.
- Playing music or computer games during class in such a way that they are visible or audible to other class members.
- Exhibiting erratic or irrational behavior.
- Behavior that distracts the class from the subject matter or discussion.
- Making physical or verbal threats to a faculty member, teaching assistant, or class member.
- Refusal to comply with faculty or teaching assistant direction.

Proper Academic Conduct: Feel free to discuss all assignments with the instructors or the TAs. Code (projects and assignment): you may discuss code solutions with other students. However:

1. You may not look at or share code with others;
2. If you discuss a solution with anyone, you must document their names in your assignment.
3. You must document this in your code if you use external resources (e.g., Piazza, StackOverflow.com, ChatGPT, or other LLMs).

Make sure that your computer account is properly protected. Use an appropriate password, and do not give your friends access to your account or computer system. Do not leave printouts, computers, or thumb drives around a laboratory where others might access them.

Programming projects will be checked by software designed to detect collaboration. This software is extremely effective and has withstood repeated reviews by the campus judicial processes.

Upon the first documented occurrence of inappropriate collaborative work or of taking a solution from a network resource, the instructors will report the academic misconduct to the Campus Judicial

Coordinator. The procedure is documented in the University of Oklahoma Academic Misconduct Code (http://integrity.ou.edu). The provider and receiver of a solution will be treated equally in the misconduct process.

University Policies

Mental Health Support Services:
Support is available for any student experiencing mental health issues that are impacting their academic success. Students can either been seen at the University Counseling Center (UCC) located on the second floor of Goddard Health Center or receive 24/7/365 crisis support from a licensed mental health provider through TELUS Health. To schedule an appointment or receive more information about mental health resources at OU please call the UCC at 405-325-2911 or visit University Counseling Center. The UCC is located at 620 Elm Ave., Room 201, Norman, OK 73019.

Title IX Resources and Reporting Requirement
The University of Oklahoma faculty are committed to creating a safe learning environment for all members of our community, free from gender and sex-based discrimination, including sexual harassment, domestic and dating violence, sexual assault, and stalking, in accordance with Title IX. There are resources available to those impacted, including: speaking with someone confidentially about your options, medical attention, counseling, reporting, academic support, and safety plans. If you have (or someone you know has) experienced any form of sex or gender-based discrimination or violence and wish to speak with someone confidentially, please contact OU Advocates (available 24/7 at 405-615-0013) or University Counseling Center (M-F 8 a.m. to 5 p.m. at 405-325-2911). Because the University of Oklahoma is committed to the safety of you and other students, and because of our Title IX obligations, I, as well as other faculty, Graduate Assistants, and Teaching Assistants, are mandatory reporters. This means that we are obligated to report gender-based violence that has been disclosed to us to the Institutional Equity Office. This means that we are obligated to report gender-based violence that has been disclosed to us to the Institutional Equity Office. This includes disclosures that occur in: class discussion, writing assignments, discussion boards, emails and during Student/Office Hours. You may also choose to report directly to the Institutional Equity Office. After a report is filed, the Title IX Coordinator will reach out to provide resources, support, and information and the reported information will remain private. For more information regarding the University's Title IX Grievance procedures, reporting, or support measures, please visit Institutional Equity Office at 405-325-3546.

Reasonable Accommodation Policy
The University of Oklahoma (OU) is committed to the goal of achieving equal educational opportunity and full educational participation for students with disabilities. If you have already established reasonable accommodations with the Accessibility and Disability Resource Center (ADRC), please submit your semester accommodation request through the ADRC as soon as possible and contact me privately, so that we have adequate time to arrange your approved academic accommodations.
If you have not yet established services through ADRC, but have a documented disability and require accommodations, please complete ADRC's pre-registration form to begin the registration process. ADRC facilitates the interactive process that establishes reasonable accommodations for students at OU. For more information on ADRC registration procedures, please review their Register with the ADRC web page. You may also contact them at (405)325-3852 or adrc@ou.edu, or visit www.ou.edu/adrc for more information.

Note: disabilities may include, but are not limited to, mental health, chronic health, physical, vision, hearing, learning and attention disabilities, pregnancy-related. ADRC can also support students experiencing temporary medical conditions.

Religious Observance

It is the policy of the University to excuse the absences of students that result from religious observances and to reschedule examinations and additional required classwork that may fall on religious holidays, without penalty.

Persons with Disability or Special Accommodation or Accommodation for any reason:

The Accessibility and Disability Resource Center is committed to supporting students with disabilities to ensure that they can enjoy equal access to all components of their education. This includes your academics, housing, and community events. If you are experiencing a disability, a mental/medical health condition that has a significant impact on one or more life functions, you can receive accommodations to provide equal access. Possible disabilities include, but are not limited to, learning disabilities, AD(H)D, mental health, and chronic health. Additionally, we support students with temporary medical conditions (broken wrist, shoulder surgery, etc.) and pregnancy. To discuss potential accommodations, please contact the ADRC at 730 College Avenue, (ph.) 405.325.3852, or adrc@ou.edu.

Adjustments for Pregnancy/Childbirth Related Issues

Should you need modifications or adjustments to your course requirements because of documented pregnancy-related or childbirth-related issues, please contact the Accessibility and Disability Resource Center at 405/325-3852 and/or the Institutional Equity Office at 405/325-3546 as soon as possible. Also, see the Institutional Equity Office FAQ on Pregnant and Parenting Students' Rights for answers to commonly asked questions.

Final Exam Preparation Period

Pre-finals week will be defined as the seven calendar days before the first day of finals. Faculty may cover new course material throughout this week. For specific provisions of the policy please refer to OU's Final Exam Preparation Period policy.

Emergency Protocol

During an emergency, there are official university procedures that will maximize your safety.
Severe Weather: If you receive an OU Alert to seek refuge or hear a tornado siren that signals severe weather.

1. Look for severe weather refuge location maps located inside most OU buildings near the entrances.
2. Seek refuge inside a building. Do not leave one building to seek shelter in another building that you deem safer. If outside, get into the nearest building.
3. Go to the building's severe weather refuge location. If you do not know where that is, go to the lowest level possible and seek refuge in an innermost room. Avoid outside doors and windows.
4. Get in, Get Down, Cover Up
5. Wait for official notice to resume normal activities.

Additional Weather Safety Information is available through the Department of Campus Safety.

The University of Oklahoma Active Threat Guidance

The University of Oklahoma embraces a Run, Hide, Fight strategy for active threats on campus. This strategy is well known, widely accepted, and proven to save lives. To receive emergency campus alerts, be sure to update your contact information and preferences in the account settings section at one.ou.edu.

RUN: Running away from the threat is usually the best option. If it is safe to run, run as far away from the threat as possible. Call 911 when you are in a safe location and let them know from which OU campus you're calling from and location of active threat.

HIDE: If running is not practical, the next best option is to hide. Lock and barricade all doors; turn of all lights; turn down your phone's volume; search for improvised weapons; hide behind solid objects and walls; and hide yourself completely and stay quiet. Remain in place until law enforcement arrives. Be patient and remain hidden.

FIGHT: If you are unable to run or hide, the last best option is to fight. Have one or more improvised weapons with you and be prepared to attack. Attack them when they are least expecting it and hit them where it hurts most: the face (specifically eyes, nose, and ears), the throat, the diaphragm (solar plexus), and the groin.

Please save OUPD's contact information in your phone.

NORMAN campus: *For non-emergencies call (405) 325-1717. For emergencies call (405) 325-1911 or dial 911.*

TULSA campus: *For non-emergencies call (918) 660-3900. For emergencies call (918) 660-3333 or dial 911.*

Fire Alarm/General Emergency: (required)

If you receive an OU Alert that there is danger inside or near the building, or the fire alarm inside the building activates:

1. *LEAVE* the building. Do not use the elevators.
2. *KNOW* at least two building exits
3. *ASSIST* those that may need help
4. *PROCEED* to the emergency assembly area
5. *ONCE safely outside, NOTIFY first responders of anyone that may still be inside building due to mobility issues.*
6. *WAIT* for official notice before attempting to re-enter the building.

OU Fire Safety on Campus