Data Structures with AI Integration

Course Description

This course covers the fundamental concepts of data structures and algorithms, emphasizing their design, analysis, and implementation. Students will learn to analyze algorithmic complexity (O-notation) and select appropriate data structures for solving complex problems. **Distinctively, this course integrates AI-driven coding assistants (e.g., GitHub Copilot, ChatGPT) as tools for code generation, debugging, refactoring, and understanding complex structures.**

Learning Objectives

By the end of this course, students will be able to:

1. **Understand** fundamental data structures (lists, stacks, queues, trees, graphs, hash tables).
2. **Analyze** the time and space complexity of algorithms.
3. **Implement** these structures in a high-level language as Python.
4. **Use AI tools** to generate boilerplate code, explain complex algorithms, and debug code efficiently.
5. **Evaluate** AI-generated code for correctness, efficiency, and security vulnerabilities.

Weekly Schedule

Part 1: Foundations & AI Pair Programming

Week	Topic	AI Integration Focus
1	**Introduction & Complexity**	Setting up AI IDE extensions. Using AI to explain complexity classes (O(n) vs O(n^2)).
2	**Functions, Arrays, and Pointers**	AI prompt engineering to generate node classes and basic traversal functions.
3	**Strings, Structs, and Memory Allocation**	Using AI to simulate real-world scenarios (e.g., CPU scheduling) using these structures.

Part 2: Advanced Data Structures & Optimization

Week	Topic	AI Integration Focus
4	**Algorithm analysis and complexity notations**	Asking AI to analyze the algorithms and identify the problems when hallucinations occur.

Week	Topic	AI Integration Focus
5	**Fundamental data structures : Linked Lists, Stacks and Queues**	Working with AI for visualizing Linked Lists, Stacks and Queues and identify differences
6	**Recursion & Search**	Asking AI to visualize recursive calls and identify base cases to prevent stack overflow.
7	**Sorting Algorithms**	AI-assisted comparative analysis of quicksort vs. mergesort.

Part 3: Hierarchical & Network Data

Week	Topic	AI Integration Focus
8	**Trees and Binary Search Trees (BST)**	AI-assisted implementation of insertion, deletion, and balancing logic of trees.
9	**Heaps & Priority Queues**	Using AI to implement priority queues for industrial engineering optimization problems.
10	**Graph Algorithms (BFS/DFS)**	Asking AI to generate traversal paths and translate between adjacency matrices and adjacency lists.
11	**Hashing & Hash Tables**	Using AI to suggest hash functions and analyze collision rates.

Part 4: Practical Application & Ethics

Week	Topic	AI Integration Focus
12	**Refactoring & Optimization**	Feeding working, inefficient code to AI and asking for optimized alternatives.
13	**Debugging AI Code**	Intentionally using flawed AI code and teaching students how to debug it.
14	**AI Ethics**	Discussion on AI hallucinations and technical debt.

AI Use & Academic Integrity Policy

1. Philosophy: The "Augmented Engineer"

In this course, AI is treated as a **sophisticated but fallible intern.** It is an essential tool for modern engineering, but it lacks true logic and accountability. Your goal is to manage the AI, not be replaced by it. You are ultimately responsible for every semicolon and every complexity claim in your submissions.

2. Permitted AI Use (The "Green Light")

You are encouraged to use AI (ChatGPT, Claude, GitHub Copilot, etc.) for:

- **Scaffolding:** Generating boilerplate code for standard data structures (e.g., a basic Linked List class).
- **Conceptual Explanation:** Asking for analogies to understand complex topics like "Graph Traversal" or "Heapsort."
- **Debugging:** Pasting your error messages to find syntax mistakes or logical bottlenecks.
- **Applications:** Brainstorming how a specific data structure can be applied to industrial optimization or supply chain problems.

3. Prohibited AI Use (The "Red Light")

The following actions constitute academic dishonesty:

- **Blind Copy-Pasting:** Submitting AI-generated code that you cannot explain line-by-line during a "viva" (oral check).
- **Unattributed Use:** Failing to include a **Prompt Log** with your assignments.
- **Exam Deception:** Using AI during closed-book midterm, final or quizzes unless explicitly stated otherwise.
- **Logic Outsourcing:** Asking AI to solve the "core logic" of an assignment without first attempting a pseudocode draft yourself.

4. The "Accountability Clause" (Crucial)

AI is known to **"hallucinate"**—it will confidently provide code that looks correct but fails on edge cases or uses inefficient algorithms (when is required).

- **The "Hallucination" Defense:** "The AI told me it was correct" is **not** a valid excuse for broken code.
- If the AI provides an incorrect solution and you submit it, you will be graded on the incorrect logic, not the AI's "intention."

5. Documentation & Transparency

Every assignment submission must include an **AI Disclosure Statement**. You must choose one of the following for every major task:

1. **Direct:** "I did not use AI for this assignment."
2. **Assisted:** "I used AI to generate boilerplate and debug. (See attached Prompt Log)."
3. **Collaborative:** "I used AI to optimize my logic and compare data structure efficiencies. (See attached Prompt Log)."

The Prompt Log Requirement: A prompt log is a simple PDF or text file showing your conversation history with the AI. It serves as your "work cited" page.

6. The "Explain Your Code" (EYC) Rule

The instructor reserves the right to call any student for a 5-minute **EYC session**. If you cannot explain the logic of the code you submitted (even the parts generated by AI), your grade for that assignment will be reset to zero. **To use the tool, you must master the tool.**

Assessment Methods

1. **AI-Integrated Assignments and Lab (40%):** Lab and Homework submissions must include the prompts used to generate code snippets and a brief reflection on how the AI helped or hindered the solution.
2. **Midterm Exam (20%):** Traditional closed-book exam to ensure fundamental understanding of concepts without AI reliance.
3. **Final Exam (30%):** Traditional closed-book exam to ensure fundamental understanding of concepts without AI reliance.
4. **Participation & AI Ethics Discussion (10%):** Engagement in discussions regarding the reliability of AI-generated algorithms.
