CS2: Object-Oriented Programming and Data Structures

COMP 2150, Fall 2025

Lecture: Monday & Wednesday, 2:20 PM – 3:45 PM
Lab: Tuesday (choose one 2-hour session; see Canvas for your assigned time)
Location: Fogelman Clsrm Bldg 127 (Lecture); Lab location Dunn Hall 232
Course Website: Canvas
Instructor: Brodrick Stigall
Email: blstgall@memphis.edu
Office Hours and Location: [TBA]

Catalog Description

COMP 2150 - CS2: OOP and Data Structures (4)
Principles of object-oriented programming and software development; problem solving with recursion and abstract data types, including linked lists, stacks, queues, binary search trees, hash tables; basic GUIs.
Prerequisite: MATH 1910 or MATH 1421 (or MATH 1830 for COMP minors) and COMP 1900.
This is not an introductory programming course! Python competency is expected before taking COMP 2150.

Course Outcomes

- Write object-oriented programs that combine functions and data.
- Write code that reads from or writes to a file.
- Write code that demonstrates effective use of error handling techniques, e.g., exceptions.
- Read and write code to manipulate data structures including linked lists, stacks, queues, trees, heaps, priority queues, sets, maps, and hash tables.
- Read and write code that uses recursion.
- Apply an object-oriented programming language to develop software including programs utilizing multiple classes.
- Analyze a problem statement to develop a mental model of objects necessary to translate abstract concepts into classes.

Required Equipment

- **Laptop:** Students must bring a working laptop to all lectures and labs.
- **Development Environment:** Students are responsible for maintaining a working Python development environment.

Required Textbook

- **zyBooks:** 2150: Object Oriented Programming and Data Structures
- zyBook code: **MEMPHISCOMP2150StigallFall2025**
- Subscribe at learn.zybooks.com
- Subscription: $89 (Available Aug 1, 2025 – Dec 15, 2025)

Evaluation

Category	**Weight**
Attendance/Class Participation	10%
In-class/Lab Assignments	10%
zyBooks Assignments	15%
Homework Assignments	15%
Quizzes (3 @ 10% each)	30%
Final Exam	20%

Letter grades will be assigned according to the standard scale, with possible downward adjustment at the instructor's discretion.

Assignments

- **In-class assignments:** Short comprehension and code-along exercises during lecture.
- **zyBooks assignments:** Complete all Participation, Challenge, and Lab activities from assigned zyBooks sections.
- **Homework assignments:** More complex coding exercises, often building on lab work.
- **Lab assignments:** Weekly hands-on programming tasks during your scheduled lab session, often using zyLabs. Labs reinforce lecture topics and provide practical experience with data structures, OOP, and debugging.

All assignments and due dates will be posted in Canvas.

Quizzes & Exams

- **Quiz 1:** Searching and sorting, recursion, basic OOP.
- **Quiz 2:** Inheritance, linked lists, stacks, queues.
- **Quiz 3:** Trees, heaps, priority queues.
- **Final Exam:** Comprehensive.

Academic Integrity

All University and course-specific academic integrity policies apply.
AI Tools: See below for detailed policy.

AI Tools (e.g., ChatGPT, Github Copilot) Policy

- **Quizzes and Final Exam:** All AI tools are prohibited.
- **Assignments/Labs:** AI tools may be used for syntax help, debugging tips, or general programming questions, but not for generating or completing solutions.
  - If you use an AI tool, you must include a statement in your submission with:
    - The name of the tool used
    - A transcript or export of your interaction with the tool
- If in doubt, ask the instructor before using an AI tool.

Classroom Behavior, Equity, Inclusion, Accommodations, and Mental Health

All University policies apply. See Canvas for full statements and resources.

Tentative Course Schedule (Fall 2025)

Week	Dates	Monday (Lecture)	Tuesday (Lab)	Wednesday (Lecture)	Notes/Holidays
1	Aug 25–27	Course intro, OOP review	Lab orientation, environment setup	OOP review, intro to course tools	First Day: Aug 25
2	Sep 1–3	**No class** (Labor Day)	Diagnostic Lab	Recursion (first introduction)	Labor Day: Sep 1
3	Sep 8–10	Recursion continued, problem solving	Recursion lab	Sorting algorithms	
4	Sep 15–17	OOP, classes	Sorting lab	Inheritance	
5	Sep 22–24	Inheritance, polymorphism	OOP/class design lab	Exception handling	
6	Sep 29–Oct 1	File I/O, exceptions	Inheritance lab	Linked lists	
7	Oct 6–8	Linked lists	File I/O lab	Stacks, queues	
8	Oct 13–15	**No class** (Fall Break)	**No lab** (Fall Break)	Stacks, queues	Fall Break: Oct 11–14
9	Oct 20–22	Stacks, queues continued	Linked list lab	Review, Quiz 1	

10	Oct 27–29	Trees, BSTs	Stack/queue lab	Tree traversals	
11	Nov 3–5	AVL trees, heaps	BST lab	Priority queues	
12	Nov 10–12	Hash tables	Heap lab	Sets, maps	
13	Nov 17–19	Sets, maps	Hash table lab	Graphs	
14	Nov 24–26	Graphs	Set/map lab	**No class** (Thanksgiving)	Thanksgiving: Nov 26–30
15	Dec 1–3	Capstone project intro	Graph lab	Capstone project presentations, review	Last Day: Dec 3
	Dec 4	Study Day: Dec 4			
	Dec 10	**Final Exam:** Wed, Dec 10, 1:00–3:00 PM			

Note: Schedule and topics are subject to change. All updates will be posted on Canvas.

Additional Notes

- **Lab Attendance:** Attendance at your assigned lab session is required. Labs are hands-on and essential for mastering course material.
- **Lab Structure:** Labs will use zyLabs for coding assignments, with undergraduate lab leaders providing guidance and support. Labs reinforce lecture topics and provide opportunities for collaborative problem-solving and debugging.
- **Makeups:** There are no makeups for in-class or lab assignments except for documented emergencies.

For full policies, resources, and updates, see Canvas.
Welcome to CS2!