YOUNGSTOWN
STATE
UNIVERSITY

CSIS 3700: Data Structures and Objects

CRN: 41053 (3 cr.)
Fall Semester 2025; Start Date 08/25/2025 – **End Date**: 12/13/2025
Modality: In-Person, Meeting Times: Mon & Wed, 10:00am – 11:15am, Location: Meshel #338

Contact Information

Professor: Hailong Jiang
Office: Meshel Hall #320
Phone: n/a
Email: hjiang@ysu.edu
Office Hours: 14:00 – 15: 00 PM Mon & Wed,12:00 PM – 2:00 PM Friday, or virtual by appointment.

Catalog Description

Program design, style and expression, testing and debugging for larger programs. Introductory concepts of object-oriented programming, including classes, methods, encapsulation, and abstract data types. Theory and application of data structures, including linked structures, priority queues, trees, networks, and graphs. Coreqs: CSIS 3700L. Prerequisites: "C" or better in either CSIS 2605 or CSIS 2610.

Course Materials

- Textbook: Data Structures and Other Objects Using C++, 4th Edition, Pearson, by Main and Savitch. ISBN-13: 978-0132129480. ISBN-10: 0132129485.
- See Computer lab locations on campus.

Course Learning Outcomes/Objectives

1. Given a problem, determine the appropriate data structures required for an efficient solution.
2. Given a non-recursive algorithm or function, determine its big-O run time.
3. Describe and implement the basic container structures — dictionaries, stacks, queues, lists, heaps, trees and graphs.
4. Useseparatecompilationandmodularprogrammingtoimplementsoftwaresolutionstoproblems

ABET Computer Science Student Outcomes Addressed by this Course

1. CS1: Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions
2. CS2: Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program ’ s discipline
3. CS5: Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.

4. CS6: Apply computer science theory and software development fundamentals to produce computing-based solutions

Attendance Expectations

Attendance is required; with few exceptions, those who do best in courses are those who attend regularly. You are allowed three unexcused absences; additional unexcused absences will resultin a 5% deduction from your overall grade. Late arrivals and early departures (10 minutes or more) count as an unexcused absence.

See YSU Attendance Policy which addresses excused absences for participation in university-sponsored events, government-related activities, religious observances, death of a family member, and documented personal illness.

Assignments/Assessments

You will be given

- 4 programming projects,
- 2 midterm exams,
- a final exam.

These assignments are due by the end of the day (midnight) on the due date as indicated in the assignment.

Late Work

An assignment is late if it is not submitted by the end of the day (2400h) on the due date. You must allocate enough time to complete the assignment on time.

For each calendar day late, 10% of the assignment's possible score will be deducted.

There is no make-up for exams. Missing an exam will greatly affect your grade since it carries a lot of weight. In case of illness or abnormal circumstances, please consult with the instructor in advance if possible to make alternate arrangements. You must formally inform the instructor in writing and present proper supporting documents within a week of the exam.

Grading and Grading Scale

Letter grades will be based on the weighted average score according to the following scale:

- 4 Programming Projects (10% each): 40%
- 2 Midterms Exams (18% each): 36%
- Final Exam: 24%

Grades are normally assigned using the following traditional 90% – 80% – 70% – 60% cutoffs. I reserve the right tolower these thresholds, but not to raise them.

Weighted Average	**Letter Grade**
90%-100%	A
80%-89%	B
70%-79%	C

60%-69%	D
< 59%	F

Please see the YSU Grading System, which includes information about grading options, withdrawals, and repetition of courses.

University Policies

You are welcome to copy and paste required university policies into your syllabus. However, you may consider using the language below and linking to policies. Linking will allow you to not have to update your syllabus should policies change. **Note: Only link to policies if you are sharing your syllabus in an online format.**

University policies can be found online and provide you guidance on your rights as a student in this course. The links below take you directly to a specific policy. Should you have any questions about a policy, please do not hesitate to contact me using the information at the top of the syllabus.

- Statement of Non-Discrimination from the University
- Academic Integrity/Honesty
- Student Accessibility
- Incomplete Grade Policy
- YSU Attendance Policy

Generative AI Usage Policy

Generative AI tools (e.g., ChatGPT, GitHub Copilot, Google Gemini) may be used in this course **only under the following conditions**:

1. **Transparency** – If you use a generative AI tool to assist with code, writing, or problem-solving, you must clearly indicate which tool you used, what prompts you provided, and how you used the generated output in your submission.
2. **Academic Integrity** – You remain fully responsible for the correctness, originality, and academic integrity of all work submitted. Copying AI-generated content without understanding or attribution will be treated as plagiarism.
3. **Learning Priority** – Generative AI should supplement, not replace, your own understanding. Over-reliance on AI tools may negatively impact your performance on exams and practical assessments.
4. **Prohibited Uses** – Generative AI may not be used during exams or for any assignments explicitly designated as “AI-free” by the instructor.

Tentative Course Schedule

The course schedule, policies, procedures, and assignments in this course are subject to change in the event of extenuating circumstances, by mutual agreement, and/or to ensure better learning.

Week	Dates	Lecture Topics (Chapters)	Lab	Project / Exam
Week 1	Aug 25–Aug 29	Ch.1–2: Introduction to Data Structures; Classes in C++	Lab 1: Introduction to Unix	
Week 2	Sep 1–Sep 5	Ch.2–3: Member Functions; Operator Overloading; Container Classes (Sep 1 - No Class, Labor Day)	Lab 2: Makefile Exercise	~~Project 1 Assigned~~
Week 3	Sep 8–Sep 12	Ch.4: Pointers, Dynamic Memory, Dynamic Arrays	Lab 3: Debugger Skills	Project 1 Assigned
Week 4	Sep 15–Sep 19	Ch.5: Linked Lists Basics	Lab 4: Linked List Practice	~~Project 1 Due~~
Week 5	Sep 22–Sep 26	Ch.5: More Linked Lists (Doubly, Circular, Sentinel Nodes)	Lab 5: Linked List Review	Project 1 Due
Week 6	Sep 29–Oct 3	Ch.6: Templates: Functions and Classes	Lab 6: Building Templates	Project 2 Assigned
Week 7	Oct 6–Oct 10	Ch.7: STL, Iterators, Generic Algorithms	Lab 7: Using the STL	
Week 8	Oct 13–Oct 17	Ch.8: Stacks, Queues, Priority Queues, Expression Evaluation	Lab 8: Using Vectors	Midterm Exam 1
Week 9	Oct 20–Oct 24	Ch.9: Recursion, Introduction to Trees	Lab 9: Recursion Practice	Project 2 Due
Week 10	Oct 27–Oct 31	Ch.10: Binary Trees: Nodes, Traversals, BST	Lab 10: Tree Traversals	Project 3 Assigned
Week 11	Nov 3–Nov 7	Ch.11: B-Trees: Constructors, Printing, Searching	Lab 11: B-Tree Project Work	
Week 12	Nov 10–Nov 14	Ch.11: Copy Constructor, Destructor, B-Tree Insertion (Nov 11 - No Class, Veterans Day)	Lab 12: B-Tree Project Work	

Week 13	Nov 17–Nov 21	Ch.11: B-Tree Removal, Heaps, Time Analysis	Lab 13: B-Tree Project Work	Midterm Exam 2
Week 14	Nov 24–Nov 28	Ch.13: Sorting: Quadratic Sorting, Quicksort (Nov 27–28 - No Class, Thanksgiving Break)	Lab 14: Sorting Practice	Project 3 Due
Week 15	Dec 1–Dec 5	Ch.12: Mergesort, Hash Tables	Lab 15: Sorting Program	Project 4 Assigned
Finals Week	Dec 8–Dec 12	Final Exam (Cumulative: Ch.1–13)		Final Exam; Project 4 Due