CS 2337 – Computer Science II (C++) Syllabus

Term	Spring 2025			
Section	CS 2337.501	ECSS 2.415	TR	0530-0645p

Instructor

Name	Prof. Gordon Arnold
Office	ECSS 4.232
Email	gordon.arnold@utdallas.edu
Hours	MW 1000-1100a or by appointment
Website	http://golear.org

Contacting the Instructor

Email: I teach a lot of classes, a lot of sections in those classes, and a lot of students. **Contacting me by email is always the best way, but you must include your class and section in the subject line of any e-mail you send.** If you do not, it will be marked as spam (not the gelatinous canned meat product) and will not be answered.

Making an Appointment: Often, making an appointment is the best way to meet with me. Please use MS Teams or Outlook to send me a meeting request. As with e-mail please include your class and section in the request.

Phone/Teams Chat: If you try to contact me by phone or through chat on MS Teams, I **cannot** guarantee a timely response.

Course Information

Course Description

Further applications of programming techniques, introducing the fundamental concepts of data structures and algorithms. Topics include recursion, fundamental data structures (including stacks, queues, linked lists, hash tables, trees, and graphs), and algorithmic analysis. Includes comprehensive programming projects. Programming language of choice is C++. Credit cannot be received for both CS 2337 and CS/CE 2336. (3-0) S

Prerequisites

AP score of at least 4.

Corequisites

(CE 2305 or CS 2305) with a grade of C or better
or (Data Science major and MATH 3315).

Learning Outcomes

- ✓ Ability to create a comprehensive programming project
- ✓ Ability to implement recursive algorithms
- ✓ Ability to implement linked lists, stacks, and queues
- ✓ Ability to implement a binary tree
- ✓ Ability to use hash tables and graphs
- ✓ Ability to understand algorithmic analysis
- ✓ Ability to implement and use generics/templates

Required Materials

- **Textbook:** *Starting Out with C++ from Control Structures to Objects, 9th ed.* (or higher) by Tony Gaddis (10th ed. is $10.99/month online)

- **zyBooks**: ZyLabs is mandatory. Labs, in class activities, assignments and practice problems will be posted in Zylabs and you will post the solutions there.

It is an auto grader tool. The feedback on your code will however be provided to you by the graders.

- **C++ Compiler and IDE:**
  - You may use any IDE you wish as long as it supports the proper version of the compiler (gcc)
  - I encourage everyone to use Online GDB (https://www.onlinegdb.com)
    - This will simulate the ZyBooks environment for testing.
    - It provides a clean debugger interface that will make it easier to fix bugs in your code.
    - Use of a debugger is one of the most important skills you should learn in this class
    - It's free and little set-up required.
  - All projects you submit will be compiled with g++ 5.4.1 with C++11 standards. You may use any IDE that can utilize at least g++ 5.4.1. Using a newer version of C++ should not create any problems as long as the proper standard is being used.
- **Note Taking Supplies**: I expect you to actively take notes. Anything discussed in class or available in the textbook is fair game on a test. Have something available to collect all the knowledge given to you (be that in physical or digital form).
- **Computer:** You will need a computer for this class. Minimum specs can be found here: https://oit.utdallas.edu/technologyinitiative/hardware. However, if you do not have a computer, the CS computer lab is open 24/7 and contains all the software you will need for this class.
  - **Use of smart phones and headphones/earphones is strictly prohibited. Laptops need to be closed, until I request to open for an in-class activity**. Tablets that use a digital pen (no keyboards or mice) are allowed for students to take notes. This is not a popular rule among students, but it is done so to maintain fairness in the academic environment for all students. There is no need to copy my examples in class as I will make them available to you. I would rather you focus on how we are solving the problem.

Grading Criteria

Assignments/Projects	40%	A+	≥97	C-	70-72
Exam #1 (Midterm)	30%	A	93-96	D+	67-69
Exam #2 (Final)	30%	A-	90-92	D	63-66
		B+	87-89	D-	60-62
Make-Up Exams	NONE	B	83-86	F	<60
Late Work	NONE	B-	80-82		
Extra Credit	NONE	C+	77-79		
Curve of ANY Kind	NONE	C	73-76		

The exams will be held in the testing center. You should register for the exams within the first two weeks of class (by Census Day).

- Exam #1 (Midterm) Thursday, March 13, 2025 (tentative)
- Exam #2 (Final) Thursday, May 8, 2025 (tentative)

You need to score an **average of at least 70% (C-) across both exams** to qualify for a grade of C or better in the course. E.g.: if your final score is a B- but your exam avg score is D, your final score will still be a C-.

Course Policies & Classroom Expectations

Comet Creed	"As a Comet, I pledge honesty, integrity, and service in all that I do."
Attendance Policy	This course is scheduled to be taught in the Traditional (in person) Mode. Regular and punctual class attendance is expected. Students who fail to attend class regularly are inviting scholastic difficulty. Historically students who skip class, don't complete or don't put much effort into their programming assignments, or get a lot of help from classmates/mentors/etc., do not perform well on exam questions testing the material covered by the assignment.
CSMC	The Computer Science Mentoring Center (CSMC) is a free resource available to all students taking this class. The CSMC provides assistance in many areas including: • Understanding core concepts related to this class • Developing a logical framework for a program • Connecting programming constructs to the logic of the program • Assisting in solving syntax and logical errors in your code • Exam reviews and reworks (by faculty request) The mentors will meet with you 1-on-1 to address your specific problem areas. Their goal is to help you understand what is wrong and how to fix it, but they will not do the work for you. For more information about the CSMC, including location and hours of operation, please visit http://csmc.utdallas.edu.
Computers, Phones, and Recording Devices	If you do not have a computer, the CS computer lab is open 24/7 and contains all the software you will need for this class. **Use of smart phones and headphones/earphones is strictly prohibited. Laptops need to be closed, until I request to open for an in-class activity.** Tablets that use a digital pen (no keyboards or mice) are allowed for students to take notes. This is not a popular rule among students, but it is done so to maintain fairness in the academic environment for all students. There is no need to copy my examples in class as I will make them available to you. I would rather you focus on how we are solving the problem.
Cheating	Homework, labs, projects, and exams are individual endeavors. I have a zero-tolerance policy for cheating. Should I discover cheating issues, I will **immediately report to the OCSC <u>without</u> warning** for disciplinary action. Cheating can include (but is not limited to) the following: • Copying answers from a classmate • Turning in answer(s) that were clearly derived from a classmate's answers • Work (and especially code) created by AI and not by you • Faking code functionality (making it pass tests without actually doing the work) The only resources you are allowed to use for your work are: • your instructor

	• the assigned textbooks (except for exams) • the CSMC (except for exams) **Students should avoid using web sites like YouTube or GitHub for help on assignments.** First, the material you find online will be arranged and taught differently. This can lead to unnecessary confusion. Second, copying code from a web site is considered plagiarism and will be treated as such. If you find code on a web site, it is highly likely another student will find it as well which may cause both submissions to be flagged for similarity. Submitting project and assignment details to web sites for outside help is also considered academic dishonesty by UTD.
Grade Disputes	All grade disputes must be reported within 1 week of the grade in question being posted in eLearning. Uncontested grades will become final after 1 week and cannot be disputed later. If you have questions regarding your grades, please contact your instructor. Please note that due to FERPA, grades cannot be discussed via e-mail.
Class Materials & Recordings	The instructor may provide class materials that will be made available to all students registered for this class as they are intended to supplement the classroom experience. These materials may be downloaded during the course, however, these materials are for registered students' use only. Classroom materials may not be reproduced or shared with those not in class, or uploaded to other online environments except to implement an approved Office of Student AccessAbility accommodation. Unless the Office of Student AccessAbility has approved the student to record the instruction, students are expressly prohibited from recording any part of this course. This includes cell phone cameras/video. **Failure to comply with these University requirements is a violation of the Student Code of Conduct, and will negatively impact your grade.**

Additional Resources

Academic Support	Please go to http://go.utdallas.edu/academic-support-resources.
Syllabus Policies	Please go to http://go.utdallas.edu/syllabus-policies for these policies.

Testing

The written portion of all exams will be done in the UTD Testing Center. All students must reserve a time slot no later than 48 hours prior to exam time at:

https://ets.utdallas.edu/testing-center/students/

If you do not register to take the exam at least 48 hours in advance, you will not be allowed to take the exam and will be **given a grade of 0**. It is recommended that you submit your registration for all three exams on the first day of class and write down the date and time of each.

Course Schedule (Tentative)

#	Week		Topic	Chapter
0	Jan 21	Jan 23	Class Intro, Introduction to C++	1-6
1	Jan 28	Jan 30	Streams, Branching, Loops, Functions	1-6, 20
2	Feb 4	Feb 6	Arrays, Searching and Sorting	8
3	Feb 11	Feb 13	Pointers and Memory Management	9
4	Feb 18	Feb 20	C Strings	10
5	Feb 25	Feb 27	Structs, Enumerations, Unions, Bit Flags	11, K
6	Mar 4	Mar 6	Classes, Constructors, Exceptions	13, 16
7	Mar 11	**Mar 13**	**Exam #1 (Midterm)**	
	Mar 18	Mar 20	**Spring Break**	
8	Mar 25	Mar 27	Copy constructor, Operator Overloading	14
9	Apr 1	Apr 3	Inheritance, Polymorphism, Pure Virtual	15
10	Apr 8	Apr 10	Linked Lists, Stacks, and Queues	18-19
11	Apr 15	Apr 17	Trees and Graphs	21
12	Apr 22	Apr 24	Hash Tables	
13	Apr 29	May 1	Hash Tables (cont.)	
14	May 6	**May 8**	**Exam #2 (Final)**	

Note: All labs and assignments are due on Saturday of that week at 11:59:00pm.

The descriptions and timelines contained in this syllabus are subject to change at the discretion of the instructor.