CS 262 – Introduction to Low-level Programming (Fall 2025)
Department of Computer Science
George Mason University

COURSE INFORMATION

Course Number and Title:	CS 465 – Computer Systems Architecture
Section Number and CRN:	Section 003 – CRN 77930 Section 006 – CRN 80173
Class Duration:	August 25, 2025, to December 17, 2025
Class Meeting Time:	Section 003 – 10:30am to 11:45am on Tuesday and Thursday Section 006 – 9:00am to 10:15am on Tuesday and Thursday
Class Location:	Section 003 – Horizon Hall Room 2014, Fairfax Campus Section 006 – Innovation Hall Room 204, Fairfax Campus

***Important Notice*:** Course syllabus, gradebook, exercises, quizzes, labs, and projects will be hosted on Canvas. If the campus closes, or if a class meeting needs to be canceled or adjusted due to weather or other concern, students should check Canvas for updates on how to continue learning.

GENERAL INFORMATION

Class Instructor:	Angkul Kongmunvattana, Ph.D. (Dr. K)
E-mail address:	akongmun@gmu.edu (start your “Subject” line with [CS262])
Office Location:	ENGR 4428, Fairfax Campus
Office Hours:	Walk-ins (2pm to 3pm on Monday) and by appointments (must be scheduled and confirmed via GMU email at least one day in advance)

TEXT AND RESOURCES

Recommended Textbooks:

- Kernighan and Ritchie, *The C Programming Language*, 2nd Edition, Prentice Hall, 1988.
- Byron S. Gottfried, *Programming with C*, 2nd Edition, Schumm's Outline, 1996.
- Peter Printz and Tony Crawford, *C in a Nutshell: A Desktop Quick Reference*, 1st Edition, O'Reilly, 2006.
- David Griffiths and Dawn Griffiths, *Head First C*, 1st Edition, O'Reilly, 2012.

COURSE CATALOG DESCRIPTION

Most high-level programming languages (and particularly Java) insulate the programmer from the realities of the hardware on which the programs will run. C is the exception since it was originally designed to implement the Unix operating system. C offers the programmer direct access to much of the underlying hardware and, for programs running under Unix, direct access to operating system services. For these reasons C remains the language of choice for systems programming.

This course is intended to prepare students for topics in systems programming. It emphasizes relevant concepts of the C programming language, as well as the use of main commands of the Unix Operating System. This is a course on "low-level" programming using C which is taught with an emphasis on operations with pointers.

Credit hours: 3
Prerequisite: CS 211 (or CS 222) AND CS 110 with a grade of C or higher.

COURSE OUTCOMES

- Be able to implement, test and debug a designed solution to a problem in a low-level programming language, specifically the C programming language.

- Demonstrate a good understanding of C language constructs such as pointers, dynamic memory management, and address arithmetic.
- Demonstrate a good understanding of C libraries for input and output, and the interface between C programs and the UNIX operating system.
- Demonstrate an ability to use UNIX tools for program development and debugging.

TOPICS

C Types, Operators, and Expressions
Basic I/O, Input and Output Libraries
Control Flow
Functions and Program Structure
Strings
Pointers and Arrays
Dynamic memory allocation
Structures
The Unix System Interface
vi/vim
Bitwise operations
Makefiles, Compiling, and Linking
Debugging using gdb and Valgrind
File I/O
Multiple source files

EVALUATION METHODS

Lab Assignments	10%	(11 labs, lowest lab dropped, each lab worth 1%)
Projects	30%	(3 programming assignments, each assignment worth 10%)
Exercises	4%	
Quizzes	16%	(4 quizzes, each quiz worth 4%)
Midterm Exam	15%	
Final Exam	25%	(cumulative)

PASSING REQUIREMENTS

The department policy stipulates that you must meet at least one of the following criteria to pass the course: (1) The average of your midterm and final exam scores must be $\geqslant$ 65. OR (2) Your final exam score must be $\geqslant$ 60.

COURSE GRADING

Grade	Cut-Off	Grade	Cut-Off	Grade	Cut-Off	Grade	Cut-Off
A+	98%	B+	88%	C+	78%	D	60%
A	92%	B	82%	C	72%	F	0%
A-	90%	B-	80%	C-	70%		

COURSE CONTENT AND CALENDAR (TENTATIVE)

Session	Date	Topic
1	August 26	Class Administration and Overview
2	August 28	C Types, Operators, and Expressions
3	September 2	User I/O and Makefile
4	September 4	Control Flow
5	September 9	Functions

6	September 11	Arrays
7	September 16	Strings
8	September 18	Pointers
9	September 23	Pointers
10	September 25	Pointers
11	September 30	Multidimensional Arrays
12	October 2	File I/O
13	October 7	Midterm Review
14	October 9	Midterm Exam
15	October 14	Midterm Recap
16	October 16	Preprocessing Directives
17	October 21	Debugging C Programs
18	October 23	Structures
19	October 28	Self-referential Structures
20	October 30	Linked List
	November 4	Election Day – No Classes
21	November 6	Valgrind
22	November 11	Bitwise Operations
23	November 13	Handling Multiple Source Files
24	November 18	Unix System Calls
25	November 20	Unix System Calls
26	November 25	Using Make to Build C Programs
	November 27	Thanksgiving Recess – University Closed
27	December 2	Linking
28	December 4	Final Review
29	December 11 (7:30am to 10:15am)	Section 006 - Final Examination (Cumulative)
	December 16 (10:30am to 1:15pm)	Section 003 – Final Examination (Cumulative)

COURSE POLICIES

Coding/Programming Policies

No sharing or discussion of codes. Unless specifically stated otherwise, all assignments are individual projects, not group projects. Students are expected to do their own work, not to share programs with each other, nor copy programs from anyone else. This means you may not discuss program design or strategize solutions with anyone except your instructor or a course UTA or GTA. However, you may offer more limited assistance to your fellow students regarding specific questions on their programming assignments by responding to queries on Piazza. Any sharing of code or discussion of programming projects, except within the parameters of Piazza, constitutes an honor code violation. Suspected honor code violations are taken very seriously and will be reported to the Honor Committee. Read the GMU Honor Code and CS Department Honor Code. You are bound by these codes.

No incorporation of code from any source external to the course. You may not incorporate code written by others, such as code found on the Internet or any of the numerous CS books available. You may freely use any code provided as part of the project specifications, without any need for crediting the source. However, if you use code provided by your instructor (other than that given as part of the project specifications) or from the course textbook, you must document what portion came from those sources.

Piazza: We encourage the use of Piazza to discuss assignments and assist one another with programming questions. You may ask questions or respond to queries on Piazza regarding projects or other assignments, so long as you do not post any C code or detailed pseudocode, and so long as you do not provide specific solutions to the overall problem or algorithm design (even in English). Often, students believe that "simple" code is acceptable to place on Piazza. However, because there is a wide variation in what different students and instructors regard as "simple," we must be very strict about the ban against this. Specifically, only an instructor, GTA or UTA is permitted to place code on Piazza unless it is the code that has already been provided to all students (either as part of the assignment specification itself or within the class textbook).

Discussing Piazza postings outside of Piazza. Please note that Piazza assistance must remain on the forum. "Summarizing" Piazza statements or responses to another student verbally regarding an assignment are *not* acceptable and are subject to the above ban on discussing assignment solutions. While it may seem harmless, Piazza was set up so that all assistance could be overseen by instructors/GTAs/UTAs, and it is nearly impossible to truly duplicate Piazza discussion outside of the actual forum, thus creating the potential for either (unknowing) mistaken advice, or for unfair advantage by certain students. If you truly wish to assist a fellow student, encourage him or her to log onto Piazza, and direct him/her to specific postings you find helpful.

Back up your program regularly. You are expected to back up your program in separate files as you get different pieces working. Failure to do this may result in a much lower grade on your submission if problems occur at the very last minute (e.g., accidentally deleting your program, having problems connecting, etc.) These problems will not be accepted as excuses.

Keep an untouched copy of your final code submission. It is important that you do not touch your programs once you have made your final submission. If there are any submission problems, consideration for credit will only be given if it can be verified that the programs were not changed after being submitted.

Code must be compiled using GNU C compiler (gcc) on Zeus and run on Zeus with no warnings and errors. Students may develop programs using any computer system they have available. However, that submitted labs and projects must be compiled under the GNU C compiler (gcc) available on Zeus. Your documentation should clearly state which software was used for compilation, and once makefiles are introduced, a makefile should be included with each assignment submission. No extensions will be given due to compiler incompatibility.

AI/LLM Policies

You are not allowed to use code from the Internet. This includes Stackoverflow, Chegg, Github, ReplIt, TutorialPoint, or any other site with code.

You are not allowed to use code from Generative AI or LLMs. All work submitted in this course must be your own original work. The use of AI writing/programming tools, such as Copilot, ChatGPT, Gemini, Claude, Grok, and other similar models are prohibited in this course and will be considered a violation of academic integrity. The use of AI-enhanced IDEs, such as VisualStudio Code, Cursor, Augment, Aider, Windsurf, Cline, or any other editor that has AI-features enabled will be considered a violation of academic integrity. You can disable AI features in IDEs and then use those IDEs without any issues. As an example, if you uninstall Copilot and select "Hide AI Features" in VS Code, you may use that IDE.

You are not allowed to discuss any implementation (code) or design decisions with any other student. This includes prior students and students from other sections.

You are not allowed to share or use any code from any other student (including past students). This includes posting your code to a public GitHub repository.

You need to understand/apply fundamental concepts and skills without the use of AI tools. Practicing without the use of these tools is critical to the process of learning.

All suspected academic integrity violations are required to be reported to the office of Academic Standards.

Lab Policy

A short programming assignment will be given on Canvas at the beginning of each lab. **Check your lab section now and make sure to attend the correct one.** The lab instructor (GTA) and one or more UTAs will be available to help students with the lab assignment. If not completed, the lab may be taken home. Labs are due at the designated for submission on Canvas. No late submissions will be accepted. One lowest lab score will be dropped. A grade of zero will be assigned for the missed lab without an excused absence (e.g., illness, unforeseen emergency, etc.). If the class instructor (not the lab instructor, GTA) deems the absence is excused, then the final exam grade will also be used for the missed lab.

Project Policy

In addition to the labs, there will be three larger programming assignments. Projects will be posted on Canvas when assigned and announced in class along with their due dates. If your project is incomplete, you may still submit it for possible partial credits, but your codes must compile and run on Zeus without obvious errors (even if all required functionality is not present). Your GTAs rely on running your codes as a part of your grade determination. All submissions must be made on Canvas. Email submissions are NOT accepted. Students are responsible for verifying that all submissions contain the correct files and that the submitted files can be extracted correctly on Zeus. Note that once a submission link on Canvas expires NO resubmissions due to corrupt or incorrect files will be accepted.

Students begin the semester with THREE (3) One-Day-Late tokens for the projects. If you have a token left and you turn in your project late, the token(s) is spent, and no late penalty will be applied. **The latest you can turn in the project is 48 hours after its due date.**

Exercise Policy

Exercises are given on Canvas, covering materials on each topic. Exercises are designed to check and to reinforce the learning of materials. Exercises can be taken multiple times. A score from the highest attempt will be recorded. Exercises cannot be completed after the last day of class (12/04/2025).

Quiz Policy

Quizzes are also given on Canvas and can be attempted only once. There are no make-up quizzes. A quiz is assigned when substantial topics have been covered in class and practiced through exercises. Each quiz is 30 to 60 minutes in duration, depending on the complexity of the topics. When assigned, quizzes are released at 6:00pm on Tuesday with a due date at 6pm on Sunday of the same week. While quizzes will remain open after their due dates for studies and reviews, late quizzes will get a zero grade. A grade of zero will be assigned for the missed quiz without an excused absence (e.g., illness, unforeseen emergency, etc.). If the instructor deems the absence is excused, then the final exam grade will also be used for the missed quiz.

Midterm Exam Policy

Midterm exam is in-person using pencil/pen and paper. It is scheduled for Thursday October 9, 2025, during the class period. A grade of zero will be assigned for the missed exam without an excused absence (e.g., illness, unforeseen emergency, etc.). If the instructor deems the absence is excused, then the final exam grade will also be used for the missed midterm exam.

Final Exam Policy

Final exam is also in-person using pencil/pen and paper. It covers all materials. The Registrar Office has scheduled our final exam for Section 003 on Tuesday December 16th, 2025, from 10:30am to 1:15pm and for Section 006 on Thursday December 11, 2025, from 7:30am to 10:15am. According to the University Policy (AP.3.10.1), absences from final exam will not be excused by the instructor except for sickness on the day of the exam.

Grade Contesting Policy

You have one week after grades are released to contact the instructor (for Exercises, Quizzes, and Exams) and the GTA (for labs and projects) with a grading issue. After this one-week period, no grades will be changed.

There are no extra-credit exercises, quizzes, labs, assignments and/or projects.

COURSE POLICY ADDENDUM

Students are expected to refer to the Common Course Policies for the following information. Students will be held responsible for knowing this information.

- Academic Standards
- Accommodation for Students with Disabilities
- FERPA and Use of GMU Email Addresses for Course Communication
- Title IX Resources and Required Reporting

CLASS COMMUNICATIONS

CS 262 will be using Piazza and Canvas for most class communications. You are responsible for any notifications or information posted on Canvas/Piazza either by your class instructor, GTA, or UTA. You will need to check Canvas and Piazza regularly for such notices. Some information may be disseminated through these channels rather than in class. Individual communications with the instructor/GTA/UTA may be done on Piazza via private posts.

IMPORTANT DATES

Midterm Exam	October 9, 2025, during class period
Final Exam	Section 003: December 16, 2025, 10:30am to 1:15pm
	Section 006: December 11, 2025, 7:30am to 10:15am

See Fall 2025 Academic Calendar for other important dates.

The syllabus may be adjusted if deemed necessary by the instructor.