

Syllabus: CS240-03 Programming in C (Fall 2026)

CLASS LOCATION/TIME

- Tue/Thu: 4:00PM-5:15PM @ HLL-3507 Auditorium

CLASS WEBSITE: All information about the class will be posted on this website

- <https://www.cs.umb.edu/~duc/www/teaching/cs240/>

INSTRUCTOR: (Note: when you email me, please always cc the TA)

- Prof. Duc Tran (duc.tran@umb.edu, McCormack M3-0201-18), office hours: Tue/Thu 2:30PM-4PM
- Teaching assistant (TA): Duy Le (duy.le004@umb.edu, McCormack M03-156), office hours: Mon/Wed 12PM-4PM

My email to the class will be sent from the WISER system, so make sure that your UMB email address is set up correctly with WISER. You should also visit the website regularly

Topics: CS 240 is an introductory course, teaching you how to write, execute, and debug C-language programs. We will cover the first 7 chapters of the Kernighan & Ritchie book, and Chapter 8 if time permits. Topics include: introduction to C; types, operators, and expressions; control flow; functions and program structure; pointers and arrays; data structures; input and output.

Prerequisites: CS110 (at least C-) or CS115L

Textbook: *The C Programming Language*. Kernighan & Ritchie. 2nd edition. Prentice-Hall.

Grading: A lot of coding practice is required in this class. There will be 5 homework assignments and 5 exams. There is no final exam. Your final grade is calculated as follows:

FINAL GRADE = 20% of homework grade average + 80% of exam grade average.

- **Homework (20% of final grade):** There are 5 homework assignments, each worth 4% of the total grade. Each assignment is a coding challenge which requires writing a C program to solve a programming problem. The score for the homework submission is what you see on the Programming Website.
- **Exams (80% of final grade):** There are 5 exams, but the best 4 out of 5 will be considered for grading. Each considered exam is worth 20% of the total grade. Each exam consists of two parts:

SCORE = 40% of coding-part + 60% of interview-part.

1. **coding-part** (1 hour 15 minutes, in-class): solve a programming problem by coding on the computer and submitting the solution code to the Programming Website for autograding. You need to bring your own laptop to the class. The score for the coding part is what you see on the Programming Website.
2. **interview-part** (10-minute one-on-one Zoom interview with TA, will be video-recorded): this is an one-on-one oral interview, conducted within a week after the exam day, time to be scheduled by the TA and each student. In this interview, the student will answer questions from the TA about the student's coding-part submission. This interview is needed to make sure the student understands completely his/her coding submission. The score for the interview part is given by the Professor.

Example:

- Student A scores 100 points on the coding part (helped with AI or by someone else). During the interview part, Student A cannot explain the coding submission, earning a score of 0 point. The overall exam score is $100 * 40\% + 0 * 60\% = 40$ (points).
- Student B scores 70 points on the coding part (all the work done by herself/himself). During the interview, Student B can explain the coding submission, clearly enough to prove his/her understanding of the coding submission, even suggesting where could be wrong, thus earning a score of 80 points. The overall exam score is $70 * 40\% + 80 * 60\% = 76$ (points).
- So, in theory, a student could use AI to get 100 points on every homework and exam's coding submission, but he/she would fail the interviews (0 point on interviews), resulting in a final grade = $20\% * 100 + 80\% * 40 = 52$ points < 60 points (a failing F grade)

Threshold (%)	93	90	87	83	80	77	73	70	67	63	60	<60
Letter grade	A	A-	B+	B	B-	C+	C	C-	D+	D	D-	F

Late Homework and Makeup Policies

- No makeup is granted for students missing any homework or exam for any reason.
- Homework may be submitted late by no more than 24 hours, weekend and holiday counted. The penalty for late submission is 10% of the homework grade.

Accommodations

The University of Massachusetts Boston is committed to providing reasonable academic accommodations for all students with disabilities. This syllabus is available in alternate format upon request. Students with disabilities who need accommodations in this course must contact the instructor to discuss needed accommodations. Accommodations will be provided after the student has met with the instructor to request accommodations. Students must be registered with the Ross Center for Disability Services, CC UL 211 (617.287.7430) before requesting accommodations from the instructor. <http://www.umb.edu/academics/vpass/disability/>. After registration with the Ross Center, a student should present and discuss the accommodations with the professor. Although a student can request accommodations at any time, we recommend that students inform the professor of the need for accommodations by the end of the Drop/Add period to ensure that accommodations are available for the entirety of the course.

Academic Integrity and Code of Student Conduct

It is the expressed policy of the University that every aspect of academic life not only formal coursework situations, but all relationships and interactions connected to the educational process shall be conducted in an absolutely and uncompromisingly honest manner. The University presupposes that any submission of work for academic credit indicates that the work is the student's own and is in compliance with University policies. In cases where academic dishonesty is discovered after completion of a course or degree program, sanctions may be imposed retroactively, up to and including revocation of the degree. Any student who reasonably believes another student has committed an act of academic dishonesty should inform the course instructor of the alleged violation. These policies are spelled out in the Code of Student Conduct. Students are required to adhere to the Code of Student Conduct, including requirements for academic honesty.

Link: [UMB Code of Student Conduct](#)

[You are encouraged to visit and review the UMass website on Plagiarism: Plagiarism Prevention & Awareness: Home](#)

Course Schedule - we follow the textbook closely.

- Week 1: Chapter 1 – part 1
 - 1.1 Getting Started
 - 1.2 Variables and Arithmetic Expressions
 - 1.3 The For Statement
 - 1.4 Symbolic Constants
 - 1.5 Character Input and Output
- Week 2: Chapter 1 – part 2
 - 1.6 Arrays
 - 1.7 Functions
 - 1.8 Arguments-Call
 - 1.9 Character Arrays
 - 1.10 External Variables and Scope
- Week 3: Chapter 2 – Types, Operators, Expressions
 - 2.1 Variable Names
 - 2.2 Data Types and Sizes
 - 2.3 Constants
 - 2.4 Declarations
 - 2.5 Arithmetic Operators
 - 2.6 Relational and Logical Operators
 - 2.7 Type Conversions
 - 2.8 Increment and Decrement Operators
 - 2.9 Bitwise Operators
 - 2.10 Assignment Operators and Expressions

- 2.11 Conditional expressions
 - 2.12 Precedence and Order of Evaluation
- Week 4: Chapter 3 – Control Flow
 - 3.1 Statements and Blocks
 - 3.2 If-Else
 - 3.3 Else-If
 - 3.4 Switch
 - 3.5 Loops- While and For
 - 3.6 Loops- Do-while
 - 3.7 Break and Continue
 - 3.8 Goto and Labels
- Week 5: Chapter 4 – Functions and Program structures
 - 4.1 Basics of Functions
 - 4.2 Functions Returning Non-integers
 - 4.3 External Variables
 - 4.4 Scope Rules
 - 4.5 Header Files
 - 4.6 Static Variables
 - 4.7 Register Variables
 - 4.8 Block Structure
 - 4.9 Initialization
 - 4.10 Recursion
 - 4.11 The C Preprocessor
- Week 6, 7, 8: Chapter 5 – Pointers and Arrays
 - 5.1 Pointers and Addresses
 - 5.2 Pointers and Function Arguments
 - 5.3 Pointers and Arrays
 - 5.4 Address Arithmetic
 - 5.5 Character Pointers and Functions
 - 5.6 Pointer Arrays; Pointers to Pointers
 - 5.7 Multi-dimensional Arrays
 - 5.8 Initialization of Pointer Arrays
 - 5.9 Pointers vs. Multi-dimensional Arrays
 - 5.10 Command-line Arguments
 - 5.11 Pointers to Functions
 - 5.12 Complicated Declarations
- Week 9, 10: Chapter 6. Structures
 - 6.1 Basics of Structures
 - 6.2 Structures and Functions
 - 6.3 Arrays of Structures
 - 6.4 Pointers to Structures
 - 6.5 Self-referential Structures
 - 6.6 Table Lookup
 - 6.7 Typedef
 - 6.8 Unions
 - 6.9 Bit-fields
- Week 11, 12: Chapter 7. Input and Output
 - 7.1 Standard Input and Output
 - 7.2 Formatted Output - Printf
 - 7.3 Variable-length Argument Lists
 - 7.4 Formatted Input-Scanf
 - 7.5 File Access
 - 7.6 Error Handling-Stderr
 - 7.7 Line Input and Output
 - 7.8 Miscellaneous Functions
- Week 13, 14: Chapter 8. The UNIX System Interface
 - 8.1 File Descriptors
 - 8.2 Low Level I/O-Read and Write
 - 8.3 Open, Create, Close, Unlink
 - 8.4 Random Access- Lseek
 - 8.5 Example - An Implementation of Fopen and Getc
 - 8.6 Example - Listing Directories
- Week 15: Course review and practice