

**CS 262 Intro. to Low-level Programming  
Summer, 2020**

**Section -B01 Class Day/Time:** TR 10:30 a.m.-1:15 p.m.    **Class Location:** Online

**Instructor:** Prof. Maya Larson

**Email address:** larson@gmu.edu

**Office Location:** Blackboard Collaborate

**Office Phone:** 703-993-6432

**Office Hours:** TR after class 1:15 – 2:15

**Graduate Teaching Assistant:** Daniel McVicker

**Email address:** dmcvicke@gmu.edu

---

**Required Textbook:** Zybooks Online Course, subscription instructions are in Blackboard Announcements

**Recommended but not required:** Kernighan and Ritchie, *The C Programming Language*, 2nd ed., Prentice Hall, 1988

---

**Class 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.

**Prerequisite:** *C or better in CS 211 or CS 222 and C or better in CS 110 (CS 110 can be a co-requisite with CS 262).*

**Course Topics:** This is a course on "low-level" programming using C. We will learn C with heavy emphasis on pointer operations.

**List of topics** (In no particular order. May change without warning.):

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

Debugging using gdb and valgrind

A course schedule is located in the Course Content folder in blackboard. This schedule is tentative and subject to change.

### Course Outcomes:

1. Be able to implement, test and debug a designed solution to a problem in a low-level programming language, specifically the C programming language.
2. Demonstrate a good understanding of C language constructs such as pointers, dynamic memory management, and address arithmetic.
3. Demonstrate a good understanding of C libraries for input and output, and the interface between C programs and the UNIX operating system.
4. Demonstrate an ability to use UNIX tools for program development and debugging.

**Grading:** In addition to the labs and projects there will be a final exam. There will be no makeups on exams except under exceptional circumstances (as judged by me), and any such makeup must be arranged in advance. Grades will be computed using a weighted average of these scores with the weights:

- lab assignments 20%
- lecture activities/zybook activities 25%
- programming projects 30%
- participation/quizzes 5%
- final exam 20%

**In addition:** To receive a passing grade in this course, you must submit all projects and labs.

**Class Communications:** We will use Blackboard for most class communications. Piazza is for questions on labs, zybook activities, and projects. You are responsible for any notifications or information posted on Blackboard/Piazza either by your instructor or the class GTA. You will need to check both regularly.

Individual communications with the professor/GTA may be done by email using your GMU email account. When you email, please be sure to include your name, the class number and the topic in the subject header. (E.g.: Subject: Jim Jones / CS 262-B01 / Project 2)

**Late Assignments:** Late submissions for assignments with due dates will have a penalty of 50%. This course proceeds at a fast pace over the summer. Each lesson builds on the material from the previous lesson. If you fall behind, it will be difficult to catch up.

You should recognize that late work can cause major penalties, so start work early! If your program isn't the way you'd like it to be when the deadline is near, submit it anyway for partial credit. **In fact, submit early and often!** Blackboard permits you to retrieve and resubmit your assignment until the due date, so you may resubmit if you improve your program prior to the deadline. **Resubmissions after the deadline require approval of your TA** (s/he may already have graded your project). If you know that you wish to resubmit a new version after the deadline, it is your responsibility to notify your TA no later than the time of the deadline, so s/he will not grade the on-time submission. **No resubmissions may be made after a project has been graded.**

**Special Accommodations:** If you are a student with a disability, please see your instructor and contact the Office of Disability Services (ODS) at (703) 993-2474. All academic accommodations *must* be arranged through the ODS: <http://ods.gmu.edu>

**Honor Code Policies:** All students are expected to abide by the [GMU Honor Code](#). This policy is rigorously enforced. All class-related assignments are considered individual efforts unless explicitly expressed otherwise (in writing). Review the university honor code and present any questions regarding the policies to instructor.

Cheating on any assignment will be prosecuted and result in a notification of the Honor Committee as outlined in the GMU Honor Code. Sharing, collaboration, or looking at any code related to programming assignments that is not your own is considered cheating. See **Programming Polices** below.

The computer science department has an additional, more restrictive [CS Honor Code](#) that you are also subject to. Make sure you read and familiarize yourself with these rules.

---

## PROGRAMMING POLICIES

- (1) **No sharing or discussion of code.** 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 solution strategies with anyone except your instructor or a CS262 GTA/UTA. However, you may offer more limited assistance to your fellow students regarding specific questions on their programming assignments by responding to queries on Piazza/Blackboard. Any sharing of code or discussion of programming projects, except within the parameters of Piazza/Blackboard, 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.
- (2) **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 or GTA (other than that given as part of the project specifications) or from the course textbook, you must document what portion came from those sources.
- (3) **Blackboard/Piazza.** We encourage the use of Blackboard and/or Piazza to discuss assignments and assist one another with programming questions. You may ask questions or respond to queries on BB/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 Blackboard/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 Blackboard/Piazza code. **Only** an instructor, GTA or UTA is permitted to place code on blackboard unless it is code that has already been provided to all students (either as part of the assignment specification itself or within the class textbook).
- (4) **Discussing Blackboard/Piazza postings outside of Blackboard/Piazza.** Please note that Blackboard or Piazza assistance must remain on the forum. "Summarizing" Blackboard/Piazza statements or responses to another student verbally regarding an assignment is *not* acceptable, and is subject to the above ban on discussing assignment solutions. While it may seem harmless, Blackboard/Piazza was set up so that all assistance could be overseen by instructors/TAs/UTAs, and it is nearly impossible to truly duplicate Blackboard/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 Blackboard/Piazza, and direct him/her to specific postings you find helpful.

- (5) **Back up your work regularly.** You are expected to back up your programs in separate files as you get different pieces working. Failure to do this may result in your getting a much lower grade on a program if last minute problems occur. (Accidentally deleting your program, having problems connecting, etc., will *not* be accepted as excuses.)
- (6) **Back up your work on an independent system such as email, Blackboard submissions, Dropbox, or other cloud backup systems.** If there are any submission problems, consideration for credit will only be given if it can be verified that the program files have an independent date before the due date. Files that only exist on your laptop will not qualify because these dates can be modified.
- (7) **Code must run on Zeus using gcc.** Students may develop programs using any computer system they have available. Please note, however, that submitted projects must run under the gcc compiler (using all specified compiler options) 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 incompatibilities.
- (8) **Do NOT place your code on any computer other than zeus, your own personal device, and private postings to piazza.** Placing code on personal github repositories, the use of external (online) compilers and IDEs, and/or posting assignment specifications or your code to "tutoring" websites is forbidden. Violations of this policy may result in a GMU Honor Code violation, even after the course has ended and a grade has been given.