CS 262: Introduction to Low-Level Programming
Course Syllabus - Spring 2022
3 credits

George Mason University
Department of Computer Science

Sections: 001, 002, 003

Instructor: Prof. Ana Loreto González Hernández
Email: loreto@gmu.edu
Office Hours: Thu 3-6 pm
Room: ENGN 2707

Lectures:

Section	Days	Time	Building	Room
001	M,W	05:55 – 07:10 pm	Enterprice Hall	80
002	T,Th	01:30 – 2:45 pm	Exploratory Hall	L004
003	M,W	12:00 – 01:15 pm	Lecture Hall	1

Textbook

- Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, 2nd ed., Prentice Hall, 1988

Complementary Books:

- Byron S. Gottfried, Programming with C, 2nd ed., Schumm's Outline, 1996 or the latest
- Peter Printz and Tony Crawford, C in a Nutshell: A Desktop Quick Reference, 1st ed., O'Reilly', 2006
- David Griffiths and Dawn Griffiths, Head First C, 1st ed., O'Reilly', 2012

Course Description

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.

C is a high-level programing language that offers the programmer direct access to much of the underlying hardware and direct access to some operating system services for programs running under Unix. These features make C the preference language of choice for system programming.

Prerequisites: (CS 110* or 101*) and (CS 211 or 222) >> *May be taken concurrently

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

Course Topics

The course will cover the following topics:

- 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
- Compiling, Linking, Makefiles, using multiple source files

Evaluation and Grading

Grade Distribution

Lab assignments	20%	Lowest one drop
Projects	30%	
Quizzes	10%	
Midterm exam	15%	
Final exam	25%	

Letter Grade Distribution

Your overall course score, S, will be the sum of these points.

S >= 98	A+
S >= 92	A
S >= 90	A-
S >=88	B+
S >=82	B
S >=80	B-
S >=78	C+
S >=72	C
S >=70	C-
S >=60	D
S <60	F

Grading Elements Policies

- **Lab assignments:** Description for lab assignments will be posted on Blackboard. Submissions must be through Blackboard by the due date.

- **Projects**: Programming projects will be posted on Blackboard and student's solutions should be submitted on Blackboard by the assigned due date. If your program is incomplete, you may still submit it but your code must run without obvious errors (even if all functionality is not present). Notice your GTA relies on running your program as part of your grade determination.

  Late submissions (Labs and Projects) will be penalized 10%points per day. For instance, if a lab assignment score is 10 points and the submission is one day after due, -1 point will be deducted, and if the submission is two days after due, -2 points are deducted.

  - The cutoff for on-time submission is 11:59 pm on the due date
  - The latest you can turn in work is 48 hours after the posted deadline, after this time the submission link will be not available any more.

Each student gets TWO Emergency-Day tokens that are automatically used by submissions that are between 0-24 and 24-48 hours late to avoid points penalty. These days are applicable only for Lab assignments.

It is high recommended to attend your lab sessions, as they are intended to clarify the necessary aspects for lab and project assignments. During labs GTAs and UTAs will provide assistance and give hints to develop your programs ☺

If your program isn't the way you'd like it to be when the deadline is near, you could submit it. The system 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. The last submission is the one graded by your GTA. No resubmissions may be made after a project has been graded.

- **Quizzes:** All quizzes are ONLINE and their availiability is from 8:00 am – 8:00 pm during the posted day. They must be taken on the scheduled date/time. Passing this time, NO make-up are given unless previously arranged with the instructor. Open notes are allowed.

- **Exams** must be taken on the scheduled date/time.

The final exam is cumulative. If you perform better on the final exam than your midterm exam, the midterm grade is replaced. If you know in advance that you are unable to take an exam by the deadline posted for a valid and unavoidable reason (such as a scheduled surgery, etc.), you must notify the professor at least one week before the scheduled exam date to make arrangements for a make-up.

Per departmental policy, you must pass a significant exam threshold to receive a passing grade in this class regardless of your performance on other assignments. To receive a passing grade in this course, your Final Exam score **MUST** be >= 60%.

Important Information

- **Class Communications:** CS 262 will be using Piazza and Blackboard for most class communications. You are responsible for any notifications or information posted on Blackboard/Piazza either by your instructor, your GTA or the class UTA(s), and you will need to check the systems regularly for such notices.
  Individual communications with the professor/GTA/UTA may be done by email using your GMU email account. When you email, please try to include your name, the class number and the topic in the subject header.

  - **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).

Cheating on any assignment will be prosecuted and result in a notification of the Honor Committee as outlined in the GMU Honor Code. Any use of websites, discussing with other students, attempting to copy, share, buy, steal, or otherwise acquire a solution other than self-creating it on your own will result in an Honor Code case with recommendation to fail the course (F) plus further measures.

Programming Policies

(1) **No sharing or discussion of code for assignments**. Unless specifically stated otherwise, all assignments are individual assignments, not group assignments. Students are expected to do their own work, not to share programs with each other, nor copy programs from anyone else. However, you may offer limited assistance to your fellow students regarding questions or misunderstandings on their programming assignments. Suspected honor code violations are taken very seriously, and will be reported to the Honor Committee. (See CS Honor Code)

(2) **No incorporation of code from any source external to the course**. You may not incorporate code written by others. Of course, you may freely use any code provided as part of the project specifications, and you need not credit the source. Working something out together with an instructor or GTA will not require crediting the source.

(3) **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 your getting a much lower grade on a program if last minute problems occur. (Accidently deleting your program, having problems connecting, etc., will not be accepted as excuses.)

(4) **Keep an untouched copy of your final code submission**. It is important that you don't 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.

(5) **Code must run on Mason gcc**. Students may develop programs using any computer system they have available. However, submitted programs must run under gcc compiler available on Mason. 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.