# GMU Spring 2025

## **CS 222:** *Computer Programming for Engineers* (3 credits)

| 7:30 AM - 8:45 AM | Tue/Thu (TR) | Blueridge Hall BL129 |
|---|---|---|

Professor: Justin Wilson (jwilso86@gmu.edu)
TAs: Nguyen Phuc Be (nbe@gmu.edu) & (To Be Updated)

**Office Hours/Support**: after class, by appointment, or email using your GMU email
**Textbook**: "Problem Solving and Program Design in C," 8th edition Hanly & Koffman
**Credits**: parts of this document sourced from previous syllabi of Hal Greenwald and Jim Chen

**Description:** Introduction to C as a second programming language with emphasis on problems and language features relevant to engineers. Students are not presumed to be familiar with C prior to this class. Topics include basic data types, pointers, elementary data structures, file/output, bitwise operations, and Unix commands for compilation and debugging. Intended as terminal course for some in programming for engineers (Source URL). *Prerequisite*: C or better in CS 112

### Course Outcomes:

1. Use the C programming language to implement, test, and debug designed solutions to problems based on the requirements given.
2. Demonstrate proficiency of C language constructs such as pointers, dynamic memory management, and address arithmetic.
3. Display a good understanding of C libraries for input and output, and the interface between C programs and the UNIX operating system.
4. Gain experience developing and maintaining programs using UNIX tools.

### Course Policies:

*Class Attendance*: Students are expected to attend lecture each week and are responsible for all material covered during lecture.

*HW/Programming Assignments*: assignments will be distributed in class and/or posted on Blackboard as they are assigned, and must be submitted as stated (in class or on Blackboard) by the assigned due date. If your work is incomplete, still submit for partial credit. **Your code must run without obvious errors (even if not all requirements are met)**. **Code that does not compile/run will receive no more than 50% credit.** Remember top-down design:

- Write each step of the initial algorithm as a comment in the main function body
- Code each algorithm step in-line or as a function call, one at a time
- Save, compile, run, and test (SCRAT) for each added statement
- You should **not** write your program as a story from beginning to the end, and then compile, debug, and run! Debugging is much easier if you code and test each step as you go.

*Work Independently*: Students are expected to work independently outside of class to become familiar with GMU computer systems, to read and review all assigned materials, and to complete all homework, programming assignments, and exams. See Academic Standards.

*Reading*: Students are responsible for reading and understanding all assigned material. Use resources such as office hours, Piazza, and Blackboard for help. Also, note that some material covered in class may not be found in your textbook. If you do not understand covered material, assignments, readings, or anything at all during the course, don't hesitate to ask.

*Due Dates*: Assignments are due in the manner stated (either through Blackboard or in class).

*Late Work*: No late submissions are permitted. Submit what you have before the deadline to be eligible to receive any partial credit. If submit late, 7.5% deducted each day late.

---

**Grading Policy:** This course is graded on the Undergraduate Regular scale, defined as a letter (column 1) and + or – (column 2) in the table below based on a weighted total rounded to the nearest integer.

| | |
|---|---|
| A: 90 – 100% | 90% <= A- < 93% <= A < 97% <= A+ |
| B: 80 – (90)% | 80% <= B- < 83% <= B < 87% <= B+ |
| C: 70 – (80)% | 70% <= C- < 73% <= C < 77% <= C+ |
| D: 60 – (70)% | D |
| F: < 60% | F |

Your course grade will be a weighted total based on:

- Homework/Programming Assignments (50%)
- Midterm Exam (25%)
- Final Exam – Cumulative (25%)

The midterm and final exams will be held in class.

**Class Communication:**

In addition to in-class lecture, CS 222 will use Blackboard for class communications and Piazza as a forum for questions. Please check these platforms regularly. Individual communications with the professor or teaching assistant may also be sent by email using your GMU email account.

**Programming Policies:**

(1) **No copying or sharing of code for assignments**. 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. Any discussion or sharing of code outside GMU Computer Science Department Honor Code Policies or Academic Standards may constitutes an honor code violation. Suspected honor code violations are taken seriously, and will be reported to the Honor Committee.

Honor Code: "*To promote a stronger sense of mutual responsibility, respect, trust, and fairness among all members of the George Mason University community and with the desire for greater academic and personal achievement, we, the student members of the university community, have set forth this honor code*: **Student members of the George Mason University community pledge not to cheat, plagiarize, steal, or lie in matters related to academic work.**"

(2) **No incorporation of code from any source external to the course**. You may not incorporate code written by others nor code generated from any external source (e.g. generative AI tools, ChatGPT, etc.). Note that these tools can be used to better understand course materials and concepts. Therefore, please credit these and other sources as documentation and references when used to enhance understanding. However, they are not to be used to generate code or solutions to homework/programming assignments.

(3) **Back up your program regularly**. You are expected to backup your program in separate files or a version control system as you get different parts of functionality working. Not only is this good practice, but failure to do this may result in a lower grade if you experience last-minute problems (e.g. accidentally removing a file, connection issues). Remember, late work is not permitted and if submit late, 7.5% deducted each day late. Submit what you have before the deadline to be eligible to receive any partial credit.

(4) **Keep an untouched copy of your final code submission**. It is important that you keep an unmodified copy of your final code submission. If there are any submission issues, consideration for credit can only be given if the program is verified that it was not changed after the deadline.

(5) **Programs must be compiled, tested, and run using UNIX**. Please be prepared to develop your programs in UNIX on Mason. In the future, according to the assignment requirements, you may be allowed to use other environments with which to develop your code. However, you still must be sure it displays properly with the vi editor and compiles properly using the GNU Compiler Collection (gcc) in UNIX on the Mason cluster. Therefore, if you decide to use a different development environment, you must be familiar with transferring your files to and from the Mason system. Also, your documentation should clearly state the hardware/software used for development. Once makefiles are introduced, a makefile should be included with each assignment submission. No extensions will be given due to compiler incompatibilities.

## Tentative Class Schedule:

The following tentative class schedule represents the schedule of topics that will be covered in the weeks indicated below. Reading assignments from the textbook or from slides/handouts and homework/programming assignments may be adjusted; however, any adjustments will be clearly stated under Blackboard Announcements.

Assignments are given roughly every week. HWs due in 1 week and Programs due in 1 to 2 weeks. The assignments are either a) homework questions or b) programs to code. Your assignments will be graded and returned before the next lecture after the due date.

| **Date:**<br>**(Tue, Thu)** | **Topics include:**<br>Note: this syllabus may be subject to modification. | **Reading Assignment** |
|---|---|---|
| Week 1<br>(1/21, 1/23) | Introduction, Syllabus, Overview of C, vi editor<br>• Thu (1/23): PA #1 assigned & 1 week due Thu (1/30) | Lecture notes & Ch. 1, 2 |
| Week 2<br>(1/28, 1/30) | Compilation platforms, Top-down Design with Functions, Selection Structures, Repetition and Loops, Logical Operators, Modular Programming<br>• Thu (1/30): HW #1 assigned & 1 week due Tue (2/6) | Lecture notes & Ch. 3, 4<br>**PA1 due 1/30** |
| Week 3<br>(2/4, 2/6) | Pre/Post Increment Operators, POSIX, User-defined functions, Function Prototypes, Arrays, Pointers, Git<br>• Thu (2/6): PA #2 assigned & 2 weeks due Thu (2/20) | Lecture notes & Ch. 5, 6<br>**HW1 due 2/6** |
| Week 4<br>(2/11, 2/13) | User-defined functions, Arrays, Pointers *continued*, C structures, scope rules, Big O notation<br>• Program #2 mid check | Lecture notes & Chapter 7<br>**PA2 mid check** |
| Week 5<br>(2/18, 2/20) | Character Arrays and Strings, Text File Processing, Reference and Dereference Operators<br>• Thu (2/20): HW #2 assigned & 1 week due Thu (2/27) | Lecture notes & Chapter 8<br>**PA2 due 2/20** |
| Week 6<br>(2/25, 2/27) | The ASCII Character Set, Unix Time, Type Casting, Debugging Techniques, Text File Processing | Lecture notes<br>**HW2 due 2/27** |
| Week 7<br>(3/4, 3/6) | Scope Rules for C Variables, Midterm Review<br>• **Thu (3/6): Midterm Exam (in class)** | Lecture notes & Chapter 10 |
| Week 8<br>(~~3/11~~, ~~3/13~~) | No classes Week 8. Spring Recess Mon. Mar 10 - Sun. Mar 16 | |
| Week 9<br>(3/18, 3/20) | Structures, Unions | Lecture notes & Chapter 11 |
| Week 10<br>(3/25, 3/27) | Binary File Processing, Bitwise Operators, Array of Structures<br>• Thu (3/27): Program #3 assigned & 2 weeks due Thu (4/10) | Lecture notes & Chapter 12 |
| Week 11<br>(4/1, 4/3) | Command Line Arguments, Programming in the Large, Makefiles, Compilation of multiple files, Pointers to Structures<br>• Program #3 mid check | Lecture notes & Chapter 9<br>**PA3 mid check** |
| Week 12<br>(4/8, 4/10) | The GNU Project Debugger, File I/O, Command Line Arguments *continued*, Docker, Recursion<br>• Thu (4/10): HW #3 assigned & 1 week due Thu (4/17) | Lecture notes & Chapter 13<br>**PA3 due 4/10** |
| Week 13<br>(4/15, 4/17) | Dynamic Memory Allocation, Pointer Review, Examples and analysis, Static variables, Binary Files *continued*<br>• Thu (4/17): HW #4 assigned & due Thu (4/24) | Lecture notes & Ch. 9 *cont.*<br>**HW3 due 4/17** |
| Week 14<br>(4/22, 4/24) | Linked Lists, Stack, Queue, Recursion *continued* | Lecture notes & Ch. 13 *cont.*<br>**HW4 due 4/24** |
| Week 15<br>(4/29, 5/1) | Tree structures, Ordered Lists, Binary Search, Examples and analysis, Linked Lists *continued*, Binary Search Tree | Final Exam Review |
| Week 16<br>(~~5/6~~, ~~5/8~~) | Last day of classes Mon. May 5 so No classes Week 16 | |
| Tue, May 13<br>7:30-10:15 AM | • **Tue (5/13): Final Exam (in class, cumulative)** | Final Exam |

**Required Common Policy Addendum (link to online page) and Other Policies:**

**Course Flexibility Policies**

Once per semester, a student can ask the professor for:

- A) “Life Happens Pass” and get a 24-hour extension on 1 assignment
- B) “One Revision Pass” and allow a student to revise 1 assignment within 2 days after it is graded to get up to a 15% increase on prior grade.

**University Requirements**

- Academic Standards: https://academicstandards.gmu.edu/
- Disability Statement: https://ds.gmu.edu/

**Academic Integrity**

GMU is an Honor Code university; please see the University Catalog for a full description of the code and the honor committee process. The principle of academic integrity is taken very seriously and violations are treated gravely. What does academic integrity mean in this course? Essentially this: when you are responsible for a task, you will perform that task. Another aspect of academic integrity is the free play of ideas. Vigorous discussion and debate are encouraged in this course, with the firm expectation that all aspects of the class will be conducted with civility and respect for differing ideas, perspectives, and traditions. When in doubt (of any kind) please ask for guidance and clarification.

**GMU Email Accounts**

Students must activate their GMU email accounts to receive important University information, including messages related to this class. Your email box may be full, so check your email regularly and make sure you have enough space.

**Office of Disability Services**

If you are a student with a disability and you need academic accommodations, please see me and contact the Office of Disability Services (ODS) at 703-993-2474. All academic accommodations must be arranged through the ODS, https://ds.gmu.edu/.

**Other Useful Campus Resources:**

Writing Center: A114 Robinson Hall; (703) 993-1200; http://writingcenter.gmu.edu

University Libraries: Ask a Librarian; https://library.gmu.edu/

**Counseling and Psychological Services (CAPS):** (703) 993-2380; http://caps.gmu.edu/

**University Policies**

The University Catalog, http://catalog.gmu.edu, is the central resource for university policies affecting student, faculty, and staff conduct in university affairs. Also, an addendum is provided by the University identifying and describing relevant university policies (link to online version).

**Important Notice for LMS**: This course will be hosted on Blackboard for the Spring 2025 semester. Please ensure you are familiar with accessing and navigating this platform.

Resources and support are available at: https://lms.gmu.edu/getting-started-students/. If you have any questions, do not hesitate to reach out to me or contact the ITS Support Center for assistance.

**Campus Closure or Emergency Class Cancelation/Adjustment Policy**: If the campus closes, or if a class meeting needs to be canceled or adjusted due to weather or other concern, students should check Blackboard [or other instruction as appropriate] for updates on how to continue learning and for information about any changes to events or assignments.

**Name and pronoun use**: If you wish, please share your name and gender pronouns with me and indicate how best to address you in class and via email. You may address me as “Justin”, “Dr./Prof. Wilson” or “Mr. Wilson” in email and verbally.

**Communication Policy**:

- What is your preferred method of contact (e.g., email, LMS, MS Teams chat, other)?
  - You may email me and/or TAs directly using GMU email or via Blackboard
  - Virtual office hours will be held via Microsoft (MS) Teams upon request. I may respond to email requesting a MS Teams meeting if better to meet than reply.
- When and how quickly will you plan to respond to student queries–for instance, will you respond on weekends or after 10 pm? Will you respond within 48 hours?
  - I usually respond to student emails on the same day it is received (including weekends), but within 24 hours at the latest. TAs may respond on Piazza sooner. I am often available in the evenings after 9 PM with limited availability during day.

**Value of this course**:

- Why do people study this area of knowledge?
  - In the US labor market, 92% of jobs analyzed require digital skills (Source URL)
- How will instruction be handled: presentations, discussions, group work, labs, etc.?
  - Lecture (slides & live coding), assignment (homework & programming), textbook
- What is the value of the course to the student now and in their future careers/professions?
  - Value of CS 222 now as a student: computer and programming skills can help you be more successful in your other classes
  - Value of CS 222 in your future careers/professions (I use these all daily at work)
    - For software or data analytics, learn about code and design best practices
    - For cybersecurity, learn about memory, command line tools, and more
    - For hardware, learn about C used in robotics, operating systems, and more
    - For cloud computing, learn about VPN, remote servers, and more
- How does this particular course fit into the student’s discipline or Mason Core requirements?
  - CS 222 is intended as terminal course for some in programming for engineers
- Why is the course content arranged in this order? How do assignments help students demonstrate key knowledge?
  - We get started with command line and UNIX tools to set us up for programming in C
  - We then use C for memory, data structures, and recursion for tree structures