



**GMU Spring 2023**

**CS 531 - Computer Systems and Fundamentals of Systems Programming**

**Instructor:** Sam (Shambhu) Sainju; Email: [ssainju@gmu.edu](mailto:ssainju@gmu.edu), [ssainju1@alumni.jh.edu](mailto:ssainju1@alumni.jh.edu)

<b>Class meeting: 7:20 pm – 10 pm/Online</b>	<b>Tuesdays &amp; Thursdays</b>	<b>Online</b>
--	---------------------------------	---------------

### **Office hours by E-mail appointment**

**CS 531** provides a hands-on introduction to systems programming with an emphasis on data structures and interfacing with the UNIX operating system. While focusing on the fundamental data structures necessary for designing and implementing systems applications, we provide an introduction to the Unix Application Programming Interface (API), signals, threads, and inter-process communication (IPC). This course is instructed from a programmatic perspective using the C programming language, with special topics in both Java and Python.

The objective of this course is to focus on the principles and pragmatic methods for designing and implementing solutions to problems in Computer Science. The course will be instructed in the C programming language and the techniques discussed will be relevant for low-level or systems programming tasks. The data structures and algorithmic design aspects of the course are relevant across many different fields and are language independent. We will, however, implement several of these in C.

This course will prepare you directly for CS571 (Operating Systems) and CS555 (Networking), as well as for CS540 (Language Processors), CS550 (Database Systems), CS551 (Computer Graphics), and CS580 (AI).

### **Topics for this semester will include.**

- C Programming Language
  - Foundations, Control Flow, and Functions
  - Memory and Pointers, Address Arithmetic, Structures, Bitwise Operators
  - Text and Binary File I/O
  - Header Files
  - Static and Dynamic Memory Allocation
  - Unix API Calls
- Analysis of Algorithm Complexity



- Stacks, Queues, and Linked Lists
- Hashing, Trees, and Graphs
- Heaps and Priority Queues
- Unix Processes, Signals, and Exceptions
- Multithreading and Inter-Process Communication
- Python and Java Overviews



## Textbooks

### Required:

- *The C Programming Language, Second Edition* by [Kernighan and Ritchie](#). (ISBN 0-13-110362-8) [[Prentice Hall](#)] [[Amazon](#)]

### Recommended:

- *Advanced Programming in the UNIX Environment, 3rd Edition* by W. Richard Stevens (Author), Stephen A. Rago (ISBN-13 978-0321637734)
- *C How to Program 8/e*,  
<http://www.deitel.com/Books/C/CHowtoProgram8e/tabid/3664/Default.aspx>

### Supplemental Texts: (not required, but helpful C references)

- *UNIX Systems Programming: Communication, Concurrency and Threads, 2 Ed.* by Kay Robbins and Steve Robbins. San Antonio, Texas, Prentice Hall ISBN-10:0130424110, ISBN-13: 978013042411
- *Data Structures A Pseudocode Approach with C* by Richard F. Gilberg & Behrouz A. Forouzan 2<sup>nd</sup> Ed ISBN-13: 978-0-534-39080-8
- *Mastering Algorithms with C* by Kyle Loudon. (ISBN 1-56592-453-3).  
[[Amazon](#)][[OReilly](#)]

### Grading:

- Homework: 40 points
- Midterm Exam: 30 points
- Final Exam: 30 points

The midterm and final exams will be Take Home Test.



Grades will be assessed on the following scale:

Grade	Cut-off	Grade	Cut-off	Grade	Cut-off	Grade	Cut-Off
A	90%	B	80%	C	70%	F	0%

As a Graduate-level Course, a satisfactory grade is a **B** or higher. A passing grade is **C** or higher, however, a C is an **unsatisfactory** grade.



### Course Schedule

(Subject to adjustment)

	<u>Topics</u> include	<u>Reading</u> <i>TCPL: The C Prog. Lang. (required)</i> <i>APUE: Adv. Prog. in the UNIX Environment (recommended)</i>	<u>Assignment</u>
Week 1	Greetings, Course overview, C Programming Language overview: Variables, Expressions, Operators, Control Flow	Lecture Notes TCPL : Chapters 1, 2, 3	
Week 2	Header files, Functions, Pointers, Arrays & Strings, Typedef, Unions, Structures, C standard library	Lecture Notes TCPL: Chapters 4, 5, 6	Homework 1 assigned
Week 3	Pointers <i>cont</i> , Memory Management, Standard I/O Library, Buffering, Strings, Math, Utilities	Lecture Notes TCPL: Chapters 7, 8 APUE Chapter 5	



Week 4	Address Arithmetic, Linked Lists: Simple, Doubly, and Circular	Lecture Notes	Homework 1 due
Week 5	Linked lists continued, Recursion, Stacks and Queues	Lecture Notes	Homework 2 assigned
Week 6	Trees and Graphs	Lecture Notes	
Week 7	Homework 3 discussion <b>MidTerm Exam</b>	Lecture Notes	Homework 2 due Homework 3 assigned
Week 8	Hashing, Heaps, and Priority Queues	Lecture Notes	
Week 9	Bitwise Operators, Files, Directories, Byte Ordering (Big/Little) Endian	Lecture Notes APUE: Chapter 4	
Week 10	Process Control, Daemon Processes, Signals	Lecture Notes APUE Chapters 8,10,13	Homework 3 due Homework 4 assigned
Week 11	Interprocess Communication	A Lecture Notes PUE Chapter 15	
Week 12	Advanced IPC , POSIX Threads	Lecture Notes APUE Chapter: 11,12,17	
Week 13	POSIX Threads (continued) and Conditional Variables,	Lecture Notes APUE Chapters: 11,12	Homework 4 due
Week 14	Socket Programming in C, Java, and Python (Language Comparison and Contrast)	Lecture Notes APUE Chapters:16,17	
Week 15	Final Exam – Cumulative		



## CLASS COMMUNICATIONS

CS 531 will be using the Blackboard system for most class communications. You are responsible for any notifications or information posted on Blackboard, and you will need to check Blackboard regularly for such notices. Some information may be disseminated through Blackboard rather than in class. Individual communications with the professor 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: Sam Jones / CS 531 / assignment 2)

---

## PROGRAMMING POLICIES

(1) **No sharing or discussion 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 these guidelines constitutes an honor code violation. Suspected honor code violations are taken very seriously and will be reported to the Honor Committee.

(See <https://oai.gmu.edu/mason-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 the instructor usually will not require crediting the source.

(3) **Back up your program regularly.** You are expected to backup 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. (Accidentally 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 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.

(5) **Code must compile with Mason gcc.** Students may develop programs using any computer system they have available. Please note, however, that submitted projects must run under a C 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. No extensions will be given due to compiler incompatibilities.

---