San José State University

Science/Computer Science
CS 47, Intro to Computer Systems, Section 1, FALL 2025

Course and Contact Information

Instructor:	Sriram Rao
Office Location:	
Telephone:	
Email:	sriram.rao@sjsu.edu
Office Hours:	Fridays 12-1p (**Zoom (https://sjsu.zoom.us/j/87954063152?pwd=Ajg4nhC4hlycw3OkLLoYbosPMX6ypY.1)**) and by appointment
Class Days/Time:	Tue/Thu: 1:30-2:45p
Classroom:	DH 450
Prerequisites:	CS/MATH 42, and CS 46B (with a grade of C-" or better).

Grader and Labs Information

Labs will be published here. (https://github.com/Tiparium/CS_47_Handouts)

Grader	Nainoa-Faulkner Jackson
Email	nainoa.faulkner-jackson@sjsu.edu

Lab Hours	
Lab Location	
Office Hours	Additional hours by appointment.
Class Discord Link	**Request Invite** **(https://discord.gg/UVv4Tsru)**

Labs will require the following in order to function:

- x86/64 processor.
  (If you have an M series Mac, or any other ARM based laptop, the labs will not work.)
- Linux OS (Any x86 based distribution should work).
  You may boot natively, or use VirtualBox. WSL will not be sufficient.
- Github account (For getting the labs.)

Course Description

Instruction sets, assembly language and assemblers, linkers and loaders, data representation and manipulation, interrupts, pointers, function calls, argument passing, and basic gate-level digital logic design.

Course Learning Outcomes (CLO)

CS47 is a foundational course in learning about Computer Systems. It describes how computers operate at a fairly low level of abstraction. For example, we'll consider:

- What are the components of a computer and how do they fit together?
- How do computers do arithmetic?
- How does the code you write actually execute?
- How does a program in a high level language like C get translated into a form the machine can execute?
- How is information stored and accessed?
- How can you write code likely to execute efficiently?

Understanding these fundamentals thoroughly is absolutely essential to your future success in computer science. The material of this course is quite detailed and requires careful and diligent study.

Upon successfully completing this course, students will have strong systems foundation on how the programs we write execute on a single computer.

Course Requirements and Assignments

Please see **Syllabus (https://sjsu.instructure.com/courses/1595954/assignments/syllabus)**

The **University Policy S16-9 (http://www.sjsu.edu/senate/docs/S16-9.pdf)** , Course Syllabi (http://www.sjsu.edu/senate/docs/S16-9.pdf) requires the following language to be included in the syllabus:

“Success in this course is based on the expectation that students will spend, for each unit of credit, a minimum of 45 hours over the length of the course (normally three hours per unit per week) for instruction, preparation/studying, or course related activities, including but not limited to internships, labs, and clinical practica. Other course structures will have equivalent workload expectations as described in the syllabus.”

Grading Information

Please see **Syllabus. (https://sjsu.instructure.com/courses/1595954/assignments/syllabus)**

Classroom Protocol

This is your class. Please ask questions. Please come prepared. Do not engage in activity that may distract other students.

I do not take attendance except for the first two classes. Students not attending either of the first two classes will be dropped to make room for students on the waiting list. Attempting to get marked as present (by have someone else attend in your place or using technological deceptions) will be considered academic dishonesty and at a minimum will result in you getting dropped from the course.

We will use canvas discussion for announcements about programming labs as well as discussion topics.

Course material developed by the instructor is the intellectual property of the instructor. Students ***should not*** publicly share or upload instructor generated material for this course such as exam questions, lecture notes, hands-on exercises or homework solutions without instructor permission.

This is my first time teaching this class. The schedule listed in the syllabus is tentative and may change based on student needs. Furthermore, the programming labs are new and we will learn by doing :-).

University Policies

Per University Policy S16-9, university-wide policy information relevant to all courses, such as academic integrity, accommodations, etc. will be available on Office of Graduate and Undergraduate

Programs' **Syllabus Information web page (http://www.sjsu.edu/gup/syllabusinfo/)** at http://www.sjsu.edu/gup/syllabusinfo/" Make sure to review these policies and resources.

Cheating/Academic Dishonesty

I take issues of Academic Dishonesty very seriously. ***Do not cheat***. ***Do not share code***. If we detect cheating in a programming assignment, you will get a **0** for that lab. Repeat offense will likely lead to a **F** grade in the course.

Artificial intelligence (AI) tools like ChatGPT, Google Gemini, and GitHub Copilot are not permitted to be used as a replacement for the writing or problem-solving components of this class. SJSU's subscription to Turnitin has an AI-detection feature, and assignments that have been determined by that application or by other convincing evidence to have been written by AI in substantial fractions will receive an automatic zero. The incident will also be reported to the University as academic misconduct.

Acknowledgements

The materials for the course (such as, labs, slides) were originally developed by the authors of the textbook (Profs. Bryant and Hallaron). They have been adapted and modified by us with permission. As such please do not post any materials from this course on public websites.

Course Syllabus

Jump to Today ✎ Edit

Science/Computer Science
CS 47, Introduction to Computer Systems, FALL 2025

Required Texts/Readings

Textbook

The required text book for this class is ***Computer Systems, A Programmer's Perspective*** (https://www.amazon.com/Computer-Systems-Programmers-Perspective-3/dp/9332573905/ref=sr_1_1?crid=LD9GU29JVIIJ&dib=eyJ2IjoiMSJ9.pv3kvhQSwJK2b5HjBZ46GDupAPfIF1CN687FFpEtNRTBQcPiAKFvj8ccFofX-uonK8dmlvNCdbvhwMZyxL_F3AEn6-X53Nxv1RG8lpOmaFMbHDml7rKfMJ9mSyme_vN4V8lGzcq2AUKp-QydQbatwpBCRfcxv3F2ZwxxZD7ftjMoJw.Zs9zVmiIT5obNJZUFle7ssOwLdQbvY3LeiCWyEH-Q8g&dib_tag=se&keywords=computer+systems+a+programmer%27s+perspective&qid=1755202596&sprefix=1) by Randal E. Bryant and David O'Hallaron, Prentice Hall, 2016 (Third Edition) (https://www.amazon.com/Computer-Systems-Programmers-Perspective-3/dp/9332573905/ref=sr_1_1?crid=LD9GU29JVIIJ&dib=eyJ2IjoiMSJ9.pv3kvhQSwJK2b5HjBZ46GDupAPfIF1CN687FFpEtNRTBQcPiAKFvj8ccFofX-uonK8dmlvNCdbvhwMZyxL_F3AEn6-X53Nxv1RG8lpOmaFMbHDml7rKfMJ9mSyme_vN4V8lGzcq2AUKp-QydQbatwpBCRfcxv3F2ZwxxZD7ftjMoJw.Zs9zVmiIT5obNJZUFle7ssOwLdQbvY3LeiCWyEH-Q8g&dib_tag=se&keywords=computer+systems+a+programmer%27s+perspective&qid=1755202596&sprefix=1) . **We will refer to this book as CSAPP-3e.** The paperback book linked here costs less than the standard edition.

There is apparently also an ebook version here which you are welcome to investigate: **eBook version.** (http://www.mypearsonstore.com/bookstore/computer-systems-a-programmers-perspective-plus-mastering-0134123832)

Note that our book has quite a few errors. You can find the errata here: **book errors** (http://csapp.cs.cmu.edu/3e/errata.html) . Several times, I've been puzzled by something in the book only to find that it was wrong.

Since we will be programming some in C, I recommend that you have access to **The C Programming Language, second edition, by Brian Kernighan and Dennis Ritchie** (https://www.amazon.com/Programming-Language-2nd-Brian-Kernighan/dp/0131103628/ref=sr_1_1?crid=1B92AMP1JB4W8&dib=eyJ2IjoiMSJ9.7MU6AwhWtlZGSokpR_0dhrgvEkfVcs1qK6h9jSW5Cgh1Y9U1Jydr

oYmnTlZJGXvr1QEuVlaJ3vb60uwGPzhc0CRlY6Zn3Y7O_AW3lAakwL4GFSsRmb2c7xEJwkrtn7fVpfROqwblhb
PbZzV97eKUsuXTcRrij6mMy3oiayxeWdrf3q_dWw5lVbefaK5Y.QZEkdagku9SSShlxkEkqE_SB_XLEq7EMdAwC
1) . Java programmers should have no problem with the subset of C that we will use.

References for learning Unix and C

Here is an excellent (introductory level) **free UNIX tutorial** (https://info-ee.surrey.ac.uk/Teaching/Unix/)

Here are a few references that will help you learn C programming

- An **excellent tutorial** (https://sjsu.instructure.com/courses/1595954/files) on C by Christian Miller (UT-Austin)
- A **free tutorial** (https://www.cs.hmc.edu/~rhodes/courses/cs134/sp19/readings/pointers.pdf) on Pointers and Arrays in C
- A standard reference for the C language is **The C Programming Language** (https://www.amazon.com/Programming-Language-2nd-Brian-Kernighan/dp/0131103628/ref=sr_1_1?crid=1B92AMP1JB4W8&dib=eyJ2IjoiMSJ9.7MU6AwhWtlZGSokpR_0dhrgvEkfVcs1qK6h9jSW5Cgh1Y9U1J
oYmnTlZJGXvr1QEuVlaJ3vb60uwGPzhc0CRlY6Zn3Y7O_AW3lAakwL4GFSsRmb2c7xEJwkrtn7fVpfROqwl
PbZzV97eKUsuXTcRrij6mMy3oiayxeWdrf3q_dWw5lVbefaK5Y.QZEkdagku9SSShlxkEkqE_SB_XLEq7EMd
1) book by Brian Kernighan and Dennis Ritchie.

References for GDB

- **Beej's Quick Guide to gdb** (https://beej.us/guide/bggdb/)
- **gdb cheatsheet** (https://csapp.cs.cmu.edu/3e/docs/gdbnotes-x86-64.pdf)

Reading

To get the most out of this class, ***prior to each lecture***, students are encouraged to read the relevant chapter from the textbook.

Other technology requirements / equipment / material

Programming assignments will be a significant part of this course. We will build C programs and use run them on Linux (such as, Ubuntu 22.04). Hence, access to a computer that runs Linux (specifically, on an x86-64 machine) will be required.

Course Requirements and Assignments

I do not grade on a curve. The exams and projects measure what you are expected to have learned. There aren't many opportunities for extra credit apart from potential bonus questions on exams.

Programming Labs: We will be doing **individual** programming assignments**. Individual programming assignments are not group projects.** If students get help on assignments, even to resolve a seemingly trivial problem, it must be documented in the code with the name of the person rendering the help and a

brief description of the help provided. Extensive help on a project will result in a reduced grade. Failure to document help, or any other forms of cheating will result in a failing grade on the assignment at a minimum and may result in failure of the course. See **Integrity (http://info.sjsu.edu/static/schedules/integrity.html)** for more information. Even in open source, you cannot copy code from one open source project to another without attribution.

Programming assignments will be distributed via Github. Please create a Github account (if you don't already have one). We will use Gradescope for assignment submission and grading.

Help on Labs: Nainoa-Faulkner Jackson (course grader) will have weekly **in-person** "**lab hours (https://sjsu.instructure.com/courses/1595954)** " in to help you with the labs. Please seek his help whenever needed.

The **University Policy S16-9 (http://www.sjsu.edu/senate/docs/S16-9.pdf)**, Course Syllabi (http://www.sjsu.edu/senate/docs/S16-9.pdf) requires the following language to be included in the syllabus:

“Success in this course is based on the expectation that students will spend, for each unit of credit, a minimum of 45 hours over the length of the course (normally three hours per unit per week) for instruction, preparation/studying, or course related activities, including but not limited to internships, labs, and clinical practica. Other course structures will have equivalent workload expectations as described in the syllabus.”

Final Examination or Evaluation

This course will have a cumulative final exam given during exam week. Per **University schedule (https://www.sjsu.edu/classes/final-exam-schedule/fall-2024.php)** this is on Tuesday, Dec. 16 from 1p-3p.

There will be two in-class exams given in the semester (the last being the final exam).

Grading Information

Determination of Grades

Grades will be calculated based on the individual project grades, the two mid-semester exams, the final, discussion participation. Briefly, the weighted distribution is,

Programming Project	40%
Midterm 1	15%
Midterm 2	15%

Final	20%
Homework/Class Participation	10%

The grade distribution will be:

Percentage	**Grade**
98 and above	A+
93-97	A
90-93	A-
87-90	B+
84-87	B
80-84	B-
77-80	C+
74-77	C
70-74	C-
67-69	D+
64-67	D
60-64	D-
59 and below	F

Classroom Protocol

This is your class. Please ask questions. Please come prepared. Do not engage in activity that may distract other students.

I do not take attendance except for the first two classes. Students not attending either of the first two classes will be dropped to make room for students on the waiting list. Attempting to get marked as present (by have someone else attend in your place or using technological deceptions) will be considered academic dishonesty and at a minimum will result in you getting dropped from the course.

Course material developed by the instructor is the intellectual property of the instructor. Students ***should not*** publicly share or upload instructor generated material for this course such as exam questions, lecture notes, hands-on exercises or homework solutions without instructor permission.

This is my first time teaching this class. The schedule listed below is tentative and may change based on student needs. Furthermore, the programming project is new and we will learn by doing :-).

University Policies

Per University Policy S16-9, university-wide policy information relevant to all courses, such as academic integrity, accommodations, etc. will be available on Office of Graduate and Undergraduate Programs' **Syllabus Information web page (http://www.sjsu.edu/gup/syllabusinfo/)** at http://www.sjsu.edu/gup/syllabusinfo/" Make sure to review these policies and resources.

Course Assignments

There are two types of assignments for this course:

1. **Weekly Written Homework:** These will be assigned on Tuesday and will be due before class the following Tuesday. Answers should be submitted on canvas. Late submissions will not be accepted. These will count towards class participation. The work you submit must be your work and your work alone.
2. **Programming Labs:** (More on this below). There will be 6 labs during the course of the semester. You will be writing C programs and using tools such as gcc, gdb, strace etc. Get familiar with the toolchain :).

Programming Labs

This programming project comprises of **individual** programming assignments**. Individual programming assignments are not group projects.** The programs you submit must be your work and your work alone.

Caveats:

1. A requirement is that you **do not** publish solutions on public GitHub repos.
2. The labs are known to be demanding and debugging can be time consuming. So, please get started early for each lab.

Early submission:

- Submissions that are at least 2 or more days ***earlier*** than the deadline will get a 10% bonus.

Late Submissions:

1. Submissions that are up to a week late will be penalized 10% of the score.
2. Submissions that are up to two weeks late will be penalized 50% of the score.
3. I will not accept submissions that are more than two weeks beyond the submission date. This will be considered as a non-submission.
4. At my discretion, non-submissions will likely result in a grade penalty.

Cheating/Academic Dishonesty

I take issues of Academic Dishonesty very seriously. ***Do not cheat***. ***Do not share code***. If we detect cheating in a programming assignment, you will get a **0** for that lab. Repeat offense will likely lead to a **F** grade in the course.

Artificial intelligence (AI) tools like ChatGPT, Google Gemini, and GitHub Copilot are not permitted to be used as a replacement for the writing or problem-solving components of this class. SJSU's subscription to Gradescope has an AI-detection feature, and assignments that have been determined by that application or by other convincing evidence to have been written by AI in substantial fractions will receive an automatic zero. The incident will also be reported to the University as academic misconduct.

Preparing for the course

Start preparing for the course by doing the tutorials:

- Learn UNIX command line tools: **UNIX Tutorial (https://info-ee.surrey.ac.uk/Teaching/Unix/)**
- Learn C: **C tutorial (https://www.cs.hmc.edu/~rhodes/courses/cs134/sp19/readings/pointers.pdf)**

Course Schedule

(Tentative) Course Schedule

Week	Date	Topic/Theme	Reading(s) from CSAPP-3e
1	8/21	Overview and Amdahl's Law	1
2	8/26	Binary Representation	2.1, 2.2
	8/28	Integer Operations	2.2, 2.3

3	9/2	Integer Operations	2.2, 2.3
	9/4	Floating Point Representation	2.4
4	9/9	Assembly Programming Basics	3.1-3.4
	9/11	Arithmetic and Logic Operations	3.5
5	9/16	Control Flow	3.6
	9/18	Procedure Calls	3.7
6	9/23	Arrays, Structs, Unions	3.8, 3.9
	9/25	Buffer Overflows	3.10
7	**9/30**	**Midterm-I**	
	10/2	Memory Hierarchy	6.1-6.3
8	10/7	Cache Memories	6.4
	10/9	Cache Performance	6.5-6.7
9	10/14	Virtual Memory	9.1-9.3
	10/16	Page Tables and TLBs	9.4-9.8
10	10/21	Dynamic Memory Allocation	9.9
	10/23	Dynamic Memory Management	9.10-9.12
11	10/28	Optimizing Program Performance	5
	10/30	Linking	7.1-7.9
12	**11/4**	**Midterm-II**	
	11/6	Exceptional control flow	8.1-8.4
13	**11/11**	**Veterans Day Holiday: No Class**	
	11/13	Process Control and Signals	8.5-8.7
13	11/18	Digital Logic and Gates I	4.2

	11/20	Digital Logic and Gates II	4.2
14	11/25	Pipelining I	4.4, 4.5.1-4.5.2
	11/27	**Thanksgiving Holiday: No Class**	
15	12/2	Pipelining II	4.5.3-4.5.10
	12/4	Pipelining III	4.5.3-4.5.10
16	**12/16**	**Final Exam: 1-3p**	