CPTS 360: Systems Programming C/C++

Course Syllabus — Fall 2024

Course Credits: 4
Meeting time: Tuesday, Thursday 9:10-10:25 AM (Aug 20-Dec 06)
Classroom: Carpenter 102
Course webpage: https://wsucpts.gitbook.io/cpts360sp24/

Instructor: Monowar Hasan
Office location: EME B53
Email: monowar.hasan@wsu.edu
Phone: +509 335 8352
Office hours: Tuesday Noon-1 PM, Thursday 11 AM-noon, or by appointments

Class announcements: Canvas
Homework/lab submissions and grades: Canvas and GitHub Classroom

How to Use this Syllabus

This syllabus provides you with course-specific information and important university policies. This document should be viewed as a course overview; it is not a contract and is subject to change as the semester evolves. Changes to the syllabus will be announced on Canvas.

1 Course Overview

System programmers embody the positive essence of "system hackers" — individuals who profoundly comprehend computer and software systems across various tiers. System programming skills are essential for implementing, maintaining, and enhancing critical systems and resolving complex software issues. This course aims to equip students with the skills to craft system software, utilize system capabilities, and construct small-to-medium scale software systems.

Catalog Description. Course Prerequisite: CPTS 223 with a C or better; CPTS 260 with a C or better or EE 234 with a C or better; admitted to a major or minor in EECS or Data Analytics. Implementation of systems programs, concepts of computer operating systems; laboratory experience in using operating system facilities taught in C/C++ programming language.

2 Learning Objectives

This class contributes to the following ABET Student Outcomes (SOs):

- **SO 1:** Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions.
- **SO 2:** Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

- **SO 6:** Apply computer science theory and software development fundamentals to produce computing-based solutions.

- **SO 7:** Acquire and apply new knowledge as needed, using appropriate learning strategies.

Note: The CS ABET Accreditation is detailed here:
https://vcea.wsu.edu/undergraduatestudies/abet-accreditation/computer-science-bs-pullman/.

3 Prerequisites

Official Prerequisites. The students must complete the following courses with a grade C or better:

- CPTS 223 (Advanced Data Structures C/C++)
- CPTS 260 (Introduction to Computer Architecture) or EE 234 (Microprocessor Systems)

Development Background. The class will involve considerable programming in C language, so acquaintance with C is expected. Students should also have experience with command line interfaces, version-controlling systems (Git), and programming in a Linux environment.

4 Textbook and Reading Materials

4.1 Required Textbooks

1. Computer Systems: A Programmer's Perspective, 3rd Edition, Randal Bryant and David O'Hallaron
   ISBN: 013409266X
   Estimated cost: $155 (hardcover), $30 (paperback), available on Bookie and Amazon

2. Operating Systems: Three Easy Pieces, Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dussea
   Available online: https://pages.cs.wisc.edu/~remzi/OSTEP/
   Estimated Cost: free (digital), $26 (softcover, available on Amazon)

3. The Linux Kernel Module Programming Guide, Peter Salzman, Michael Burian, Ori Pomerantz, Bob Mottram, Jim Huang
   Available online: https://sysprog21.github.io/lkmpg/

Note: The textbooks do not cover all material discussed in class, and are not a substitute for attending lectures.

Useful C Resources

The following resources are useful to learn the C programming language.

- The C Programming Language, 2nd Edition, Brian W. Kernighan and Dennis M. Ritchie
  Available in Owen Science and Engineering Library

- The C Book, Mike Banahan, Declan Brady and Mark Doran
  Available online: https://publications.gbdirect.co.uk/c_book/

- Essential C, Nick Parlante
  Available online: http://cslibrary.stanford.edu/101/EssentialC.pdf

5 Content Outline and Schedule

The tentative weekly schedule is given below. This schedule may be revised as the course progresses. The updated schedule and class lectures will be available on course website.

Week	Topics
Week 1	Course logistics and overview, C Programming recap
Week 2	Computer memory systems
Week 3	Running programs on a system: Linking
Week 4	Control flow and process
Week 5	Process scheduling
Week 6	Kernel programming
Week 7	Week 1-Week 6 review Mid-term Exam
Week 8	Virtual memory
Week 9	System-level I/O
Week 10	File system implementation
Week 11	Network programming
Week 12	Network proxy
Week 13	Concurrent programming
Week 14	Concurrency issues and bugs
Week 15	No class (Thanksgiving break)
Week 16	Exam review, recap, and closing
Week 17	Final Exam

6 Programming Assignments

There will be Five programming assignments. The following is the tentative schedule for the assignments.

Release	Deadline	Assignment
Week 3	Week 4	Lab 0: C programming rehash — simulate a Unix file system tree *Design and implement a Unix file system tree simulator using C*
Week 5	Week 6	Lab 1: Cache memory *Simulate a cache memory system*
Week 8	Week 9	Lab 2: Simulating CPU scheduling policies *Implement FCFS, RR, and SJF scheduling policies*
Week 10	Week 11	Lab 3: Introduction to Linux kernel programming *Implement basic kernel-level Linux development functions*
Week 12	Week 13	Lab 4: Develop a web proxy *Implement a Web proxy that interacts with the browser and the rest of the web*

Lab 0 is optional for those who are confident about their C skillset. For students who submit all five labs, the **best four** scores among the five submissions will be counted for final grading. If you do not submit Lab 0, your lab programming scores in the final grade will be assigned based on Lab 1-Lab 4.

Lab 4 requires a demonstration to course staff. Details of this activity and the scheduling process for the demo session will be discussed in the class.

We will use GitHub Classroom to deliver the assignments. You will submit the assignment repository link (from GitHub) on Canvas. Detailed instructions for assignment submission will be provided on the course website. If this is your first time using GitHub, we recommend creating a GitHub account by Week 3.

7 Exam Information

The course will consist of two exams.

Week	**Exam**
Week 7	Mid-Term Exam *Includes lecture materials from Week 1-Week 7*
Week 17	Final Exam *Topics include all lecture materials and programming assignments*

All exams are closed-book and must be attempted by all students. Students can bring a handwritten cheat sheet (maximum one page, US letter size). You must write on paper in a traditional way — no digital printout (even your own handwriting in digital form from a tablet/laptop/smartphone) is allowed. Further details will be discussed before the exam. Unless otherwise specified, the exams will take place during the lecture hours programmed for this course. All exams will take place in-person.

Final Exam Information. Final exams are scheduled for the last week of the semester; the Final Exam Schedule is posted on the Registrar's website (www.registrar.wsu.edu).

8 Communication

We will communicate announcements through Canvas. Lecture materials and other learning resources will be accessible on the course website. Off-class Q&A sessions will be conducted via email/Canvas. To streamline communication, if you have questions about course materials, lectures, or project milestones, please contact the instructor via Teams/email. We recommend enabling notifications on Canvas to stay updated on important course logistics.

Canvas will be the platform for submitting class practice exercises and grading. The programming labs will be managed using a combination of GitHub Classroom and Canvas.

9 Class Participation

Students should make all reasonable efforts to attend all class meetings; however, there is **no penalty** for missing classes. While additional slides will be available online, using these materials and textbooks aims to enhance preparedness for in-class sessions, yet they cannot replace the value of attending lectures. Consequently, the course expects students to attend every class, engaging actively and constructively in discussions. Class participation will be assessed based on contributions to in-class discourse, discussions, and questions. Frequently missing class meetings may increase difficulty in understanding exam materials, homework, and programming labs.

10 Late Submission Policy

Programming Assignments. Unless otherwise noted, programming assignments will be released Tuesday by 5 AM and due by the following week Thursday at 11:59 PM (US Pacific Time). Late programming assignments **will be accepted** for up to **three** days of the deadline with a 10% reduction each day. **After three days, no submission will be graded** unless accompanied by a strong and documented reason, such as medical or family emergencies.

For example, points will be assigned based on the following strategy for an assignment with a total point of 100 and a due date of Thursday at 11:59 PM.

Submission Time	**Raw Score**	**Penalty**	**Assigned Score**
On or before Thursday 11:59 PM	α	0	α
After Thursday 11:59 PM and by Friday 11:59 PM	α	10%	α -10
After Friday 11:59 PM and by Saturday 11:59 PM	α	20%	α -20
After Saturday 11:59 PM and by Sunday 11:59 PM	α	30%	α -30
After Sunday 11:59 PM	Not graded	∞	0

Exams. Missed exams with a compelling and documented reason **may be accommodated** and will receive a grade at the instructor's discretion.

11 Grading Policy

The final course grade will be calculated using the following breakdown:

- Programming Assignments 40% (10% each from best four out of five)
- Mid-Term Exam 25%
- Final Exam 35%

We will convert the numeric scores to letter grades using the following scale mapping:

Score	**Grade**	**Score**	**Grade**	**Score**	**Grade**
≥ 90	A	[70, 75)	B-	[50, 55)	D+
[85, 90)	A-	[65, 70)	C+	[45, 50)	D
[80, 85)	B+	[60, 65)	C	< 45	F
[75, 80)	B	[55, 60)	C-		

Incomplete Grades. Academic Regulation 90 (https://registrar.wsu.edu/academic-regulations/) states that a grade of Incomplete (I) may be entered only if "the student is unable to complete their work on time due to circumstances beyond their control." Incomplete grades will be handled on a case-by-case basis at the discretion of the instructor.

12 Expectations for Student Effort

Beyond the time for lecture attendance, students are expected to invest a minimum of 5 hours outside class for each lecture equivalent (or 10 hours per week), including the time for working on programming assignments.

13 Academic Integrity

Academic integrity is the cornerstone of higher education. Therefore, all members of the university community share the responsibility of upholding and promoting the principles of integrity in all activities, including academic pursuits and honest scholarship. Academic integrity will be rigorously enforced in this course.

The students are responsible for reading WSU's Academic Integrity Policy (https://communitystandards.wsu.edu/policies-and-reporting/academic-integrity-policy/), which is based on Washington State law (https://apps.leg.wa.gov/wac/default.aspx?cite=504-26-202).

If you cheat in your work in this class, you will:

1. Receive a **-100% (i.e., Zero)** for that work (homework, lab, exam). The second offense will result in an **automatic F** grade.
2. Be reported to the Center for Community Standards (https://communitystandards.wsu.edu/).
3. Have the right to appeal the instructor's decision.
4. Not be able to drop the course or withdraw from the course until the appeals process is finished.

It will be at the discretion of the grader (if applicable) and the instructor to determine if any assignment displays evidence of collaboration exceeding these boundaries. Any effort to evade the intent of these regulations will be regarded as a breach of the essential directive. Avoid providing or seeking assistance from fellow students. If you have inquiries about permissible actions in this course, please consult the instructor.

If you want to ask for a change in the instructor's decision about academic integrity, use the the following form: https://cm.maxient.com/reportingform.php?WashingtonStateUniv&layout_id=10, available at the Center for Community Standards website (https://communitystandards.wsu.edu/). You must submit this request within 21 calendar days of the decision.

What is allowed?

- Engaging in discussions with classmates, TAs, or the instructor about potential approaches or strategies for solving homework problems or completing lab assignments is allowed.
- All homework and labs should be completed individually, including the writing of reports.

What is not allowed (and would be considered as dishonesty/plagiarism/cheating)?

- Copying (verbatim or with slight modifications) homework, code, lab reports, or any other course-related submissions from other students or online sources is prohibited.
- Disseminating or sharing one's own homework solutions, code, or other course submission material with others is prohibited. Please take measures to protect your submissions and code, both physically and digitally. You will be held responsible even if your homework or lab code is copied without your consent.
- Referring to online or Internet sources, AI tools, or textbook solution guides for completing homework or programming assignments is permissible. If you incorporate code obtained from online sources or AI tools into your lab work, it is essential to provide clear citations or acknowledgments of the original author or tool. When presenting AI-generated outcomes, ensure to include the specific prompt(s) that

led to the solution. Besides, the corresponding section of the assignment should clearly mark with the following statement: ***"This part of this assignment is completed using AI tool <name>."*** However, utilizing external resources may lead to potential grading penalties; it is advisable to consult the instructor prior to utilizing such sources.

- Using any unauthorized means to complete homework, exams, or labs is prohibited.
  Remember: If in doubt, please check with the instructor!

14 University Policy

Students are responsible for reading and understanding all university-wide policies and resources pertaining to all courses (for instance: accommodations, care resources, policies on discrimination or harassment), which can be found in the university syllabus (https://syllabus.wsu.edu/university-syllabus/).