

COSC 30203 035 - Computer System Fundamentals

Dr. XXX

Fall 2025

Note for students: The syllabus is your first course reading. It provides an orientation to, overview of the flow, and expectations of the course. You should turn to the syllabus for details on assignments and course policies.

1 Course & Instructor Information

1.1 Course

- Course Title, Prefix, Number, Section: COSC 30203 035 - Computer System Fundamentals
- Semester and Year: Fall 20XX
- Number of Credits: 3
- Course Component Type: In person lecture
- Class Location: XXX
- Class Meeting Days & Times: XXX
- Webpage: TCU Online

1.2 Instructor

- Office: TXXX
- Telephone: XXX
- Office Hours: [XXX](#)
- Preferred Method of Contact: Email
- Email: [XXX](#)
- Response Time: 24 hours during the business week. 48 hours during the weekend or holidays.

1.3 Final Evaluative Exercise and Important Dates

- Monday, September 1: Labor Day Holiday (no classes)
- Fall Break: 22:00 on Wednesday, October 8 through 8:00 on Monday, October 13.
- Last Day to Drop: Monday, November 3
- Thanksgiving Break: 22:00 on Friday, November 21 through 8:00 on Monday, December 1.
- P/NC Date: Monday, December 1.
- Study Days: Thursday and Friday, December 4-5
- Final Exam: Thursday, December 11, 11:00 - 13:30
- Final Exam Details: According to the Faculty/Staff Handbook [“Final Evaluative Exercise Policy”](#) TCU requires a “final evaluative exercise in all classes” during the designated finals period.

- **POLICY STATEMENT:** There are only two reasons for rescheduling finals:
 1. If the published final examination schedule would require a student to take more than two final examinations in a 24-hour period, the student can arrange to take one of the exams at another time. The student shall determine which final examination is to be rescheduled. The rescheduled exam shall be given at a time mutually agreeable to the student and the faculty member. A final exam may not be rescheduled so as to violate the 24-hour rule. Rescheduling arrangements must be made one week prior to the last day of classes. Unless the student is graduating, the exam must be taken during the final exam week.
 2. Students for whom a final examination conflicts with a major religious holiday or custom. Students who, in the first two weeks of the semester, notified faculty or staff of their intention to be absent from class or event, as required in the TCU Religious Holiday Policy, may reschedule a final exam within final examination week if it compromises their ability to fully celebrate a religious holiday. For example, students may reschedule late afternoon or evening examinations to that morning or at an alternate time during final examination week so they are not being tested after fasting for many daylight hours. A final exam may not be rescheduled so as to violate the 24-hour rule.

1.4 Student Resources & Policy Information

Click or scan QR code for resources to support you as a TCU student. Please note section on [Student Access and Accommodation](#) and [Academic Conduct & Course Materials Policies](#).



2 Course Description

2.1 Catalog Description

Topics include a review of machine representation of data, machine language, systems programming, memory hierarchy, linking, exceptional control flow, system-level I/O, network programming, and concurrent programming.

2.2 Prerequisites & Concurrent Enrollment

Prerequisites: COSC 20803 - Data Structures, CITE 30103 - UNIX/Linux Systems Programming, MATH 20123 - Discrete Math I; all with C- or better.

3 Course Materials

3.1 Required Materials

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Pearson, 3rd ed., 2016.

3.2 Supplemental Materials

Stephen Prata, C Primer Plus, Addison-Wesley, 6th ed., 2014.

W. Stevens, Advanced Programming in the UNIX Environment, Addison-Wesley, 3rd ed., 2013.

4 Learning Outcomes

4.1 Course Outcomes

1. Demonstrate an understanding the von Neumann model of computation, compilation systems, storage devices and hierarchy, operating system resource management concepts, and the importance of abstraction in computer systems.
2. Explain and analyze information storage (hexadecimal notation, words, and data sizes), integer representations, integer arithmetic, and floating-point representation.
3. Implement and program basic and intermediate assembly and machine language programs. This also includes being able to understand the assembly output from a compiler.
4. Explain and differentiate computer program linking systems: static and dynamic linking, symbol resolution, and relocation.
5. Explain and compare different types of exceptional control flow such as exceptions, process creation, and signals.
6. Understand and write computer programs involving Unix I/O, text and binary files, buffered and unbuffered I/O.
7. Analyze different memory systems and virtual memory. Design and implement software libraries for dynamic memory management.
8. Understand and write computer programs involving basic network programming and the client server model. Understand and write computer programs using ports and sockets in computer systems.

Course outcomes will be measured by using homework assignments, quizzes, programming projects, and exams.

5 Course Requirements

5.1 Assignments

There will be regularly assigned homework and programming contests/challenges to complete. Students will submit assignments via the appropriate dropbox created in TCU Online, GitHub Classroom, or on [Gradescope](#).

5.1.1 Quizzes & Exams

There will be regularly assigned quizzes to complete. Some quizzes will be in-class. Other quizzes can be completed outside of class. Quizzes will be administered using TCU Online.

There will be three equally weighted exams on approximately the 5th, 10th, and finals week. Proper exam etiquette will be enforced (no talking, no cheating/copying answers, no leaving the exam room to use the restroom or get a drink of water, etc).

5.2 Grading Philosophy

5.2.1 Late Work

Late homework assignments are not accepted.

Late quizzes are not accepted.

Requests for re-evaluation of points on exams, assignments, and projects must be returned to the instructor within one week, and accompanied by a brief written description of the grading error you believe was made. After this time, grades are final. Re-evaluations will not be done in the classroom, before, during, or after class. Resubmission for re-evaluation subjects the entire assignment for review. This means that if an error was made in your favor, you may lose points when re-submitting.

5.2.2 Participation, Engagement, and Attendance

Formal attendance will not be taken, however...

- **you are expected to attend lecture during our regularly scheduled class times.**

You will be considered responsible for all material presented during the lectures. Due to the nature of this course, this course will require that your mind and body show up to every lecture. As the semester progresses, I will begin to recognize students attending (or not attending) lecture. At times, lectures will cover a mathematical or theoretical analysis of algorithms. Other times lectures will cover more applied “how-to’s” and high level concepts. In addition to simply being in class, you should come prepared to ask questions about the material being covered that day.

Your participation in the course will involve the following forms of activity:

1. Attending the lectures.
2. Reading the textbook(s) and other assigned articles.
3. Completing the homework, quizzes, and projects.
4. Taking the exams.

Because it is considered an infringement on student privacy for me to have access to student medical records, I cannot accept medical documentation to justify absences. If you have a legitimate reason for your absence and want to provide verification, please access the Absence Documentation Form [here](#).

Distinction between Excused Absences and Verified Absences: Excused Absences or Official University Absences are absences described in the Official University Absence Policy and include the following: Title IX related issues, military leave, holy days, and university related absences. As faculty we may not penalize students for these absences and must allow for the completion of assignments and exams within a reasonable amount of time after the absences. Beyond these, faculty retain all discretion for consideration of a student’s absence, including verified absences.

5.3 Course Requirements & Policies

- 40% Homework, Quizzes, Programming Assignments
- 60% Exams

5.4 Undergraduate Grading Scale

Final course grades are rounded to the nearest integer prior to assigning a letter grade.

≥93	90-92	87-89	83-86	80-82	77-79	73-76	70-72	67-69	63-66	60-62	<60
A	A-	B+	B	B-	C+	C	C-	D+	D	D-	F

6 Course Schedule and Topics (always subject to change)

WEEK	DATES	TOPIC
INTRODUCTION, C PROGRAMMING TUTORIAL		
1	8/19	Overview
	8/21	C Programming I
2	8/26	C Programming II
	8/28	C Programming III
PROGRAM STRUCTURE AND EXECUTION		
3	9/2	Bits and Integers I
	9/4	Bits and Integers II
4	9/9	Machine Language - Basics
	9/11	Machine Language - Control
5	9/16	TBA Review
	9/18	Exam 1
6	9/23	Machine Language - Procedures
	9/25	Machine Language - Data
7	9/30	Machine Language - Advanced
	10/2	Memory Hierarchy
8	10/7	Cache Memories
	10/9	No Lecture - Fall Break
9	10/14	Dynamic Memory Allocation
RUNNING PROGRAMS ON A SYSTEM		
	10/16	Linking
10	10/21	TBA / Review
	10/23	Exam 2
11	10/28	ECF: Exceptions and Processes
	10/30	ECF: Signals and Nonlocal Jumps
INTERACTION AND COMMUNICATION BETWEEN PROGRAMS		
12	11/4	System I/O
	11/6	Network Programming I
13	11/11	Network Programming II
	11/13	Concurrent Programming
14	11/18	Synchronization I
	11/20	Synchronization II
	11/25	No Lecture - Thanksgiving
	11/27	No Lecture - Thanksgiving
15	12/2	TBA / Review
	12/4	No Lecture - Study Day
16	12/11	Final Exam - 11:00 - 13:30