

# **Comp 311 – Computer Organization**

**Fall 2022 Section 001 Offering**

**Instructor: Leonard McMillan**

## **Bulletin Description**

Introduction to computer organization and design. Students will be introduced to the conceptual design of a basic microprocessor, along with assembly programming. The course includes fundamental concepts such as binary numbers, binary arithmetic, and representing information as well as instructions. Students learn to program in assembly (i.e., machine) language. The course covers the fundamentals of computer hardware design, transistors and logic gates, progressing through basic combinational and sequential components, culminating in the conceptual design CPU.

## **General Course Info**

Term:	FALL 2022
Department:	COMP
Course Number:	311
Section Number:	001
Time:	TTh, 8:00 – 9:15am
Location:	Coker 201
Website:	<a href="http://www.csbio.unc.edu/mcmillan/?run=Courses.Comp311F22">http://www.csbio.unc.edu/mcmillan/?run=Courses.Comp311F22</a>

## **Instructor Info**

Name:	Prof. Leonard McMillan
Office:	SN 316
Email:	<a href="mailto:mcmillan@cs.unc.edu">mcmillan@cs.unc.edu</a>
Phone:	919-590-6078
Web:	<a href="http://www.cs.unc.edu/~mcmillan">http://www.cs.unc.edu/~mcmillan</a>
Office Hours:	W 2:00 – 4:00

## **Textbooks and Resources**

This semester I will be teaching from my notes and online materials. There are two optional books that can be used if you feel the need for supplemental materials.

Patterson & Hennessy, "Computer Organization and Design RISC-V Edition: The Hardware Software Interface" 2nd Edition, ISBN-13: 978-0128203316

Patterson & Waterman, "The RISC-V Reader: An Open Architecture Atlas," ISBN-13: 978-0999249116

## Course Description

Comp 411, Computer Organization, explores the topic of how computers work, in terms of both software and hardware. It covers a wide range of topics including what a bit is, and why bits are the atoms in the universe of computation. We also discuss how information is represented and processed in hardware, and arrive at the conclusion that, to a computer, everything is data, including the instructions that underlie software.

Comp 411 also covers the wide range of languages, and layers of translation, used for computation-- spanning from machine language to assembly language to high-level compiled and interpreted languages. We will also touch on the conventions that will enable us to construct large programs, modular software systems, and even programs that manage the loading, execution, and creation of other programs.

We will explore how simple combinational logic can be made to perform math and manipulate bits and how logic with state can be made to perform a series of operations. This will culminate in the virtual construction of a simple, yet fully functional computer.

Finally, the class considers issues of performance. What the measures of MIPS and CPI mean, and how they can be improved. We will discuss simple techniques for increasing the rates at which computers execute instructions including pipelining and parallelism. We will then address techniques for improving the apparent memory bandwidth of a computer and finally how to simulate more memory than we can actually afford.

Comp311 builds upon the foundation established in Comp211, in particular, the C programming language, data types, binary representations, and logic.

## Target Audience

This course is required for both the BS and BA computer science majors. It is also well suited for graduate students from departments outside of computer science who are interested in the inner workings of computers.

## Prerequisites

All students are required to have taken COMP 211 and all of its prerequisites (Comp 110 or equivalent, and Comp210).

## Goals and Key Learning Objectives

Comp 311 exposes the underlying mechanisms and operation of modern computers. It also exposes students to a broad range of abstractions relating to the engineering that are required to design and orchestrate large digital systems. Students will be exposed to logic, information theory, data representations, assembly language programming, and the layers of translation required to convert a program specification into a functioning program. When completed each student is expected to understand the principles underlying machine language, assembly language, ALUs, CPUs, and to have been introduced to computer system concepts including memory hierarchy, virtual memory, and processor virtualization.

## Course Requirements

In addition to lecture notes students may be assigned occasional reading supporting the topics presented in lecture. All materials will be provided on-line. Students will be assigned problem sets, with associated programming assignments to be completed on their own computers. There will be six homework assignments, two midterm exams, and a final exam. No late assignments will be accepted. However, the lowest homework score will be dropped.

## Key Dates

Midterm 1: September 22, 2022  
Midterm 2: November 1, 2022  
Final Exam: December 3, 2022 (Saturday, Darn it!, 8:00am-11:00am)

## Grading Criteria

The final grade will be based on the follow weighting factors:

6 – Problem Sets (lowest dropped)	30%
2 – In-class Exams	40%
1 – Final Exam	30%

## Course Policies

This section should address the following:

- Attendance is expected, but no roll will be taken.
- Late problem sets will not be accepted.
- Special accommodations for exams and/or make-up due to illness must be approved by the instructor.

The course final is given in compliance with UNC final exam regulations and according to the UNC Final Exam calendar.

## Honor Code

Collaboration on assigned problem sets is encouraged. However, what you hand in must be your own work.

Collaboration on tests (In-class exams and the final) is, of course, a violation of the Honor Code. This includes discussion of questions on a midterm, or final with students that have not yet taken the test, or any use of shared documents and/or social media during tests.

Using any unauthorized information sources on an exam is a violation of the honor code.

## Course Schedule

A course schedule and handouts from each lecture will be posted on the course website.

## Diversity and Inclusion Statement:

Comp 311 will attempt to provide an objective learning experience that respects all demographic groups and LGBTQ+ identifications. The instructor acknowledges that many aspects of computer science are tinged by a history that over represents a small subset of privileged voices.

Furthermore, it is possible that materials could be presented in ways that expose both overt and implicit biases of both the field and the instructor. The instructor is committed to integrating a more diverse point-of-view into this course, and always encourages feedback on how to improve the course materials to make them more inclusive and sensitive.

## Disclaimer

*"The professor reserves the right to make changes to the syllabus, including problem-set due dates and test dates. These changes will be announced as early as possible."*