

COMP-4003 Course Syllabus

1 Course Information

Course Name	COMP-4003: Computer Organization (Foundational Course)
Prerequisites	None
Required Textbook	No textbook required

2 Course Description

This course covers the overall structure of computers, and how to program a CPU. Specific topics covered in the course are:

- Components of a generic computer, and how they work together
- Motivation for binary usage, number bases (decimal and binary)
- Binary arithmetic: addition, subtraction
- Number representation in computers, signed vs. unsigned, two's complement
- Binary operations: multiplication, division, bitwise
- Other numeric representations, hexadecimal, floating point (IEEE-754)
- CPU operation details, instruction set architecture, registers and memory access
- Assembly language 1: MARIE, opcodes, arguments, registers
- Building simple programs MARIE, memory access, memory layout
- Program stack, stack implementation, and stack frames for functions
- Assembly language 2: Y86, introduction, ISA overview
- Writing/reading programs in Y86, using the built-in stack operations, function implementation
- Putting it all together, HLL to Assembly and Back Again (x86)
- (time permitting) Caching and memory hierarchy
- (time permitting) Virtual memory, paging and segmentation
- (time permitting) Pipelining in CPUs, how it works, stalls and recovery

3 Textbook and Materials

There is no required textbook for participation in this class. All materials will be available for students enrolled in the course at <https://canvas.du.edu> and online at various internal and external web sites. You will need a good internet connection and a laptop that meets DU specifications. (See <https://www.du.edu/it/support/how-to/students/laptops>). For technical support in using Canvas, please go to <http://otl.du.edu/knowledgebase/canvas>

4 Course Learning Objectives

Students that complete this course should have a strong foundational understanding of how computers work down to the CPU level. They will then be able to do the following:

1. Given a CPU ISA, be able to translate binary programs into high level assembly and understand basic program operation.
2. Given a CPU ISA, implement basic programs in high level assembly and convert them into hexadecimal or binary.
3. Understand how memory access is performed in a CPU, identify immediate, relative and offset addressing.
4. Identify stack operations and stack frames in memory, understand how values are passed to/from functional blocks using the stack and/or registers.
5. Be able to convert between arbitrary positional number systems (e.g., hexadecimal, binary and decimal)
6. Know how numbers are stored in computers, and their associated limitations and tradeoffs, such as the difference between signed and unsigned representations, and floating point.
7. Explain memory hierarchy in modern computers and the reason for various levels of memory, the different types of caches and how cache lookups are performed.
8. Understand the reasons for virtual memory, the difference between paging and segmentation, and the relative strengths and weaknesses of each.
9. Fundamentally grasp how programs are stored, loaded, and executed on a CPU.
10. Describe how a program is transformed from a text file in a high level language into binary code the a CPU can execute.

5 Course Outcomes

The high level outcomes from this course are as follows:

1. Comprehend and explain various parts of modern computers, including memory hierarchy and CPU architecture.
2. Be able to recognize and interpret low-level (assembly) programs in binary, as a foundation for more advanced topics such as: reverse engineering, disassembly and vulnerability research.

6 Program Level Goals

Courses in the cyber security MS program, including this one, should contribute to overall program level outcomes for students. This course contributes to the following program level goals:

1. Apply programming and core CS topics; including network programming, algorithms and data structures, and computer organization.
2. Understand how modern operating systems work (with a leaning towards Unix style OSes). Be familiar with their security components (identification, authentication, access controls, auditing) and how to configure them.

7 Knowledge Units

This course covers the following NSA cybersecurity Knowledge Units:

1. Operating Systems Concepts (OSC, Core Technical CDE)
2. Low-level Programming (LSP, Optional KU)
3. Systems Programming (SPG, Optional KU)

8 Attendance Policy

Regular attendance is expected. Students are responsible for all material covered in class, including announcements and handouts.

9 Academic Integrity

All work submitted for this course must be the student's own original work. Any instance of plagiarism or cheating will be dealt with according to the university's academic integrity policy. Refer to the Student Rights and Responsibilities, as well as the University of Denver Student Honor code, here: <https://studentaffairs.du.edu/student-rights-responsibilities>

10 Disability Services

If you have a disability that may affect your ability to complete the work for this course, please contact the Disability Services office at: <https://studentaffairs.du.edu/disability-services-program>