COMP 228 - System Hardware

Computer Science and Software Engineering
Fall 2025

Course Instructors:
Instructor: Hakim Mellah (hakim.mellah@concordia.ca)
Lecture time: Mon/Wed 1h30PM – 2h15PM

Office Hours:
Monday and Wednesday – 10:00AM to 11h00AM
Office: ER1129

Tutorials:
Tutorials are an integral part of the course and will be used to discuss material relevant to the assignments and projects. Tutorials will also be used to discuss solutions of the assignments. All problems related to the assignments and projects should be addressed to the tutor first.

Please see your class schedule for details

Course Objectives:
To introduce the concepts of digital logic design, computer organization and assembly language programming (Intel x86). Students will be exposed to the fundamentals of CPU hardware (central processing unit) at a level appropriate for beginner students of computer science and software engineering. At the end of the course, the student will understand how a computer is able to execute a program. A secondary objective is to gain basic knowledge of the Linux operating system and its command line interface.

Prerequisites: MATH 203 or Cegep Mathematics 103, MATH 204 or Cegep Mathematics 105

Specific Knowledge and Skills Needed for this Course:
Students taking this course are expected to have sufficient knowledge of the following topics. Should you have difficulties in any of these topics, you are strongly encouraged to review them before the DNE deadline.

- Basic mathematical concepts

Course materials
Required Textbook: N/A
Suggested Textbooks:

1. *Computer Organization and Embedded Systems, 6th edition*. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, and Naraig ManjikianMcGraw-Hill, 2012, ISBN 978-0-07-338065-0.
2. The Essentials of Computer Organization and Architecture, 6th Edition, Linda Null, Jones and Bartlett Learning, 2024, ISBN 978-1-284-25943-8.

3. Logic and Computer Design Fundamentals 3rd edition, M. Morris Mano and Charles R. Kime, Pearson Prentice Hall, 2004, ISBN 0-13-140539-X.
4. Computer Organization and Architecture Designing for Performance, 9th Edition, William Stallings, Pearson Prentice Hall, 2013, ISBN 978-0-13-293633-0.
5. A set of lecture notes will be made available in a publicly accessible directory from the Gina Cody School ENCS Linux systems (/home/t/ted/PUBLIC/COMP228). They will provide supplementary information. The course lectures will be primarily based upon the material in these lecture notes.

Here is a very nice digital logic simulator (free!) for Windows, MacOS, and Linux :

https://github.com/hneemann/Digital

Students are encouraged to download and install this digital logic simulator on their own and use it in order to more fully understand digital logic circuits.

Grading Scheme:

Assessment tool	**Weight**
Practice Assignments	**0%**
Intel Assembly Language programming assignment	**5%**
Quizzes	**10%**
Midterm examination (22 October)	**35%**
Final examination	**50%**

Additional Notes:

The **quizzes** will be based on the practice assignments. You are not required to complete the practice assignments, but you are highly encouraged to do so. The quizzes will be held during the tutorials and announced ahead of time. There will be 4 quizzes throughout the term. The NASM assembler (available on the ENCS Linux systems) will be used for the assembly language programming assignments. If you do not have an ENCS computer account, you should obtain one by visiting the ENCS Service Desk located in H964.

The **midterm** exam will be held in class (during class time) on Wednesday 22nd October.

The **final** exam will be scheduled and administered by the Exams Office during the final examination period (Dec 3 - Dec 18, 2025). The final exam will cover material from the entire semester.

There is no standard relationship between numerical percentages and the final letter grades

Course Syllabus:
The following is a tentative course syllabus. The relevant chapters of the *Computer Organization and Embedded Systems* text are given for reference purposes. The exact course content may be slightly varied from the syllabus due to time constraints:

Unit #1: Chapters 1 and 9 (selected sections) and course notes:
Number systems: integer and fractional binary numbers, hexadecimal representation. Conversion between bases. Signed number representation (sign and magnitude, one's complement, two's complement). Addition, subtraction, multiplication, division of integers. Floating point representation
of real numbers and floating-point operations. Introduction to digital logic circuits: Boolean algebra, minimization of Boolean expressions, combinational and sequential logic circuits

Unit #2: Chapter 2 and Appendix E (selected sections) and course notes:
Overview of assembly language, data movement instructions, data manipulation, addressing modes, branching, subroutines. machine instruction encoding and assembly process.

Unit #3: Chapter 5 (selected sections) and course notes:
Block diagram of a computer system (CPU, control unit, registers, main memory, address bus, control bus, data bus). The Fetch/Execute cycle, basic CPU organization: registers and buses. Control units.

Unit #4: Chapter 8 (selected sections) and course notes:
Memory systems: RAM, ROM, design of memory systems, cache memory, virtual memory.

Unit #5: Chapter 3 (selected sections) and course notes:
Input/Output organization, input/output techniques (polling and interrupts).

Notes:

1. **DO NOT** send messages using Moodle messaging, all course related questions are to be sent directly to my **hakim.mellah@concordia.ca** email address. I will be able to reply quicker to email sent to this address as opposed to replying to Moodle messages. If I do not reply within a day or two, please send a follow-up email as it is possible that your original email may have been accidentally misplaced/ deleted/not read due to the high volume of email which I receive.

2. You are advised to familiarize yourself with the Academic Code of Conduct (Academic) with regards to *plagiarism and cheating*. Refer to the appropriate sections of the Concordia University Undergraduate Calendar. All students should become familiar with the University's Code of Conduct (Academic) located at:

   https://www.concordia.ca/about/policies/spotlight/academic-code-of-conduct.html

3. Only ENCS approved electronic calculators are allowed during exams (midterm and final). The calculators must have an ENCS sticker which indicates it is an approved calculator. If the calculator is bought from the bookstore, a sticker may be affixed by the bookstore personnel upon request. If an approved calculator has been purchased elsewhere, a sticker may be obtained by visiting Carol Williams in EV.227.

Graduate Attributes:

As part of either the Computer Science or Software Engineering program curriculum, the content of this course includes material and exercises related to the teaching and evaluation of graduate attributes. Graduate attributes are skills that have been identified by the Canadian Engineering Accreditation Board (CEAB) and the Canadian Information Processing Society (CIPS) as being central to the formation of Engineers, computer scientists and information technology professionals. As such, the accreditation criteria for the Software Engineering and Computer Science programs dictate that graduate attributes are taught and evaluated as part of the courses. The following is the list of graduate attributes covered in this course, along with a description of how these attributes are incorporated in the course. The evaluation of these attributes will be based upon: assignments, midterm exam, final exam.

Table 1.CEAB Graduate Attributes and Indicators and Evaluation Method

Graduate Attribute	Indicators
Knowledge base	Levels of system abstraction and the von Neumann model. Basics of digital logic design. Data representation and manipulation. Instruction set architecture. Processor internal organization and interconnections. Assembly language programming. Memory subsystem and cache design. Input/Output subsystem. Introduction to system software.
Design	Practical use of low-level computer design principles using assembly language.
Use of engineering tools	Practical use of assembly language (Intel x86) and software tools (nasm, ld, gdb) for low-level system programming and interaction.
Individual and team work	Individual and team work on the use of assembly language for various low-level programming tasks.
Life-long learning	Locate online material and learn from it in order to complete assignments/programs.

Table 2: Learning Outcomes

Graduate Attribute	Indicators
Knowledge base	Understand the basics of digital logic design as it pertains to computer organization. Understand how computers represent and manipulate data. Understand processor internals, assembly language programming, memory design and hierarchy, I/O.
Design	Design, implement, and validate assembly language programs for solving simple problems.
Use of engineering tools	Use tools (text editor, assemble, loader, debugger) and techniques for developing assembly language programs for low-level system programming and system interaction.
Individual and team work	Deliver results of individual and team work on various low-level programming tasks/assignments.
Life-long learning	Search for and locate online reference material and learn from it for continuous self-improvement, self-learning and problem solving in the area of computer architecture and low-level system programming.

Table 2 lists the course learning outcomes which should be able to achieve by the end of the course.

A note on the use of AI tools:
The use of AI tools such as ChatGPT may enhance the learning experience, as such students may be allowed to use these tools for some purpose, as announced by the course instructor. However, any use of such tools to write assignments, programs, or any other evaluative exercise for this course **without explicit permission will not be permitted** and will be considered a violation of the Academic Code of Conduct.