湖北工業大學
HUBEI UNIVERSITY OF TECHNOLOGY

Course Title	Computer Architecture and Assembly Programming
Course Code	CMPT 3302
Semester	Summer 2026
Course Length	4 Weeks, 60 Contact Hours
Credits	4
Instructor	TBA
Office	TBA
Email	TBA
Prerequisite	CMPT 1011 Introduction to Computer Science (or equivalent introductory programming course in a high-level language) CMPT 2141 Data Structures and Algorithms

Course Description:

This course examines the interaction between hardware and software by studying how high-level programs are translated and executed at the machine level. Topics include data representation, instruction set architectures, and the execution cycle of modern processors. Students develop proficiency in assembly language programming and analyze how constructs such as control flow, procedures, and data structures are implemented in memory. The course also introduces memory hierarchy concepts and system-level interactions, providing a foundation for understanding program execution from compilation through runtime.

Course Goals:

Students who successfully complete this course will demonstrate competency in the following general education core goals:

- **Critical thinking skills** – Students will engage in creative and/or innovative thinking, and/or inquiry, analysis, evaluation, synthesis of information, organizing concepts, and constructing solutions.
- **Communication skills** – Students will demonstrate effective written, oral, and visual communication.
- **Teamwork** – Students will demonstrate the ability to work effectively with others to support a shared purpose or goal and consider different points of view.
- **Social responsibility** – Students will demonstrate intercultural competency and civic knowledge by engaging effectively in local, regional, national, and global communities.

Student Learning Outcomes:

Upon completion of this course, students will be able to:

- Represent data types (integer, floating-point) in binary and hexadecimal formats;
- Manage memory addresses and the runtime stack for subroutine execution;
- Analyze the performance impact of computer architecture components like the CPU, bus, and cache;
- Explain how pointers and memory addresses are used to implement data structures in low-level programming;
- Relate high-level language constructs (e.g., functions, arrays) to their assembly-level implementation;
- Deconstruct high-level program logic (loops, conditionals) into assembly instructions.

Textbooks/Supplies/Materials/Equipment/ Technical Requirements:

Textbooks:

- **Primary:** *Computer Organization and Design: The Hardware/Software Interface (ARM Edition)* by Patterson & Hennessy. Morgan Kaufmann, 2017.
- **Optional:** *Computer Systems: A Programmer's Perspective, 3rd edition,* by Bryant & O'Hallaron. Pearson.

Environment:

- ARMv8 assembly simulator (RARS / equivalent)
- GCC compiler
- Linux or Linux-like environment
- Debugging tools (e.g., GDB)

Course Requirements:

Programming Labs (40%)

Students complete a sequence of programming tasks involving assembly language and low-level system programming with on-site guidance and feedback from the instructor. Assignments emphasize implementing arithmetic operations, control flow, memory access, and procedure calls, as well as examining how high-level constructs (such as arrays, pointers, and functions) are realized at the machine level. Each assignment includes coding, testing, and short technical explanations.

In-Class Quizzes (10%)

Short quizzes are administered periodically during lecture sessions to assess understanding of key concepts such as instruction formats, number systems, and memory addressing. These help reinforce incremental learning and ensure continuous engagement with the material.

Midterm Examination (20%)

The midterm exam is a closed-book, written assessment conducted during a regular class meeting period. The midterm exam evaluates understanding of foundational topics including data representation, instruction execution, and assembly programming. It includes analytical problems, code tracing, and short conceptual explanations.

Final Examination (30%)

The final exam is administered in person according to the official exam schedule. The final is comprehensive and focuses on integrating knowledge of processor

organization, memory hierarchy, and system-level behavior. It includes both conceptual reasoning and applied problem-solving involving assembly and architecture.

Lab Requirements

- Labs focus on assembly programming, debugging, and system-level observation
- Students must complete experiments independently
- High-level language assignments will be exclusively in C to ensure exposure to direct memory management
- Code clarity and documentation are required
- Academic integrity policies strictly apply

Assessments: Activity	**Percent Contribution**
Programming Labs	40%
In-Class Quizzes	10%
Midterm Examination	20%
Final Examination	30%

Grading:

Final grades will be based on the sum of all possible course points as noted above.

Grade	**Percentage of available points**
A	94-100
A-	90-93
B+	87-89
B	84-86
B-	80-83
C+	77-79
C	74-76
C-	70-73
D	64-69
D-	60-63
F	0-59

Course Schedule:

The schedule of activities is subject to change at the reasonable discretion of the instructor. Minor changes will be announced in class, major ones provided in writing.

CMPT 3302 Schedule		
Lecture	Topic	Readings (*COD*)
L1	Introduction to Computer Systems & Abstraction	1.1-1.3
L2	Hardware Components & System Overview	1.4-1.5
L3	Performance and Power Considerations	1.6-1.8
L4	Binary Representation & Number Systems	2.4
	Lab 1: Binary & Data Representation Exercises	
L5	Integer Representation & Arithmetic	2.4, 3.2
L6	Instruction Set Overview (ISA Concepts)	2.1-2.3
	Lab 2: Assembly Environment Setup & Basic Instructions	
L7	Arithmetic & Logical Instructions	2.2, 2.6

L8	Control Flow & Branching	2.7
	Lab 3: Branching and Conditionals in Assembly	
L9	Loops and Program Structures in Assembly	2.7
L10	Procedures, Stack, and Function Calls	2.8
L11	Arrays, Pointers, and Memory Addressing	2.13-2.14
L12	Compilation, Assembly, and Linking Process	2.12
	Lab 4: Compilation Pipeline & Linking Exploration	
L13	Instruction Encoding & Execution Cycle	2.10
L14	**Midterm Examination**	
L15	Datapath and Processor Organization	4.1-4.4
L16	Pipelining Concepts and Design	4.5-4.6
L17	Pipeline Hazards and Performance	4.7-4.8
	Lab 5: Pipeline Hazard Visualization	
L18	Advanced Control & Exceptions	4.9
L19	Memory Hierarchy Overview	5.1-5.2
	Lab 6: Memory Access Patterns & Latency	
L20	Cache Organization and Performance	5.3-5.4
L21	Virtual Memory Concepts	5.7
	Lab 7: Virtual Memory & Address Translation	
L22	System-Level I/O and Program Interaction	2.9, 6.9
	Lab 8: System Calls and Basic I/O Programming	
L23	Linking, Runtime Behavior, and Memory Layout	2.12
L24	Parallelism and Modern Architectures (Intro)	1.8, 4.10
L25	Course Review	
---	**Final Examination**	

Accommodation Statement

Academic accommodations may be made for any student who notifies the instructor of the need for an accommodation. It is imperative that you take the initiative to bring such needs to the instructor’s attention, as he/she is not legally permitted to inquire. Students who may require assistance in emergency evacuations should contact the instructor as to the most appropriate procedures to follow.

Academic Integrity Statement

Each student is expected to maintain the highest standards of honesty and integrity in academic and professional matters. The University reserves the right to take disciplinary action, up to and including dismissal, against any student who is found guilty of academic dishonesty or otherwise fails to meet the standards. Any student judged to have engaged in academic dishonesty in coursework may receive a reduced or failing grade for the work in question and/or for the course.

Academic dishonesty includes, but is not limited to, dishonesty in quizzes, tests, or assignments; claiming credit for work not done or done by others; hindering the academic work of other students; misrepresenting academic or professional qualifications within or without the University; and nondisclosure or misrepresentation in filling out applications or other University records.

Other Items:

Attendance and Expectations

All students are required to attend every class, except in cases of illness, serious family concerns, or other major problems. We expect that students will arrive on time, be prepared to listen and participate as appropriate, and stay for the duration of a meeting rather than drift in or out casually. In short, we anticipate that students will show professors and fellow students maximum consideration by minimizing the disturbances that cause interruptions in the learning process. This means that punctuality is a must, that cellular phones be turned off, and that courtesy is the guiding principle in all exchanges among students and faculty. You will be responsible for the materials and ideas presented in the lecture.

Assignment Due Dates
All written assignments must be turned in at the time specified. Late assignments will not be accepted unless prior information has been obtained from the instructor. If you believe you have extenuating circumstances, please contact the instructor as soon as possible.

Make-Up Work
The instructor will not provide students with class information or make-up assignments/quizzes/exams missed due to an unexcused absence. Absences will be excused and assignments/quizzes/exams may be made up only with written documentation of an authorized absence. Every effort should be made to avoid scheduling appointments during class. An excused student is responsible for requesting any missed information from the instructor and setting up any necessary appointments outside of class.

Access, Special Needs and Disabilities
Please notify the instructor at the start of the semester if you have any documented disabilities, a medical issue, or any special circumstances that require attention, and the school will be happy to assist.