CS 251: Computer Architecture & Assembly Language
Fall 2021

Instructor:	Michael P. Rogers
Office:	Halsey 214
Office Hours:	MoWeThFr 3-4 PM, Tu 1:30-3 PM, Other hours by appointment
Email:	rogersm@uwosh.edu
Phone:	920-424-1388 / 309-825-6454
Class Times:	MoWeFr 11:30-12:30 PM (Mo: Halsey 101C; WeFr: Halsey 202)
Credits:	3

Prerequisites: A grade of C or better in CS 221.

Description: An introduction to RISC-based instruction set architecture. Topics include: data representation, assembly language programming, run-time storage management, pointers and references as exemplified in the C++ programming language, and introduction to system software.

Course Website: if it happens in this course, it will be posted on UWO's Canvas site. Set up notifications to be alerted when announcements are posted, new assignments, quizzes, or notes are posted and graded, etc.

Required Textbook:
ZyBooks

Subscription Instructions:

1. Sign in or create an account at learn.zybooks.com
2. Enter zyBook code: UWOSHCOMPSCI251RogersFall2021
3. Subscribe

Course Outcomes :
Upon successful completion of the course, students will be able to:

1. Express characters and integers in binary, hexadecimal, signed and unsigned representations.
2. Determine whether overflows occur in signed or unsigned additions and subtractions of integers.
3. Write normalized and denormalized floating point numbers in single and double precision using the IEEE 754 Floating Point Standard.
4. Analyze the IEEE 754 Floating Point Standard and determine what integers cannot be represented exactly by the Floating Point Standard.
5. Organize the memory layout of global integers and characters assuming the Little-Endian and Big-Endian notations.
6. Edit an assembly language program, assemble the program and print output on console using Linux.
7. Design assembly language program given high-level source code.
8. Implement assembly language programs that read in integers from console, process the input and print results on the console.
9. Implement high-level language control structures in assembly language.
10. Implement one and two-dimensional arrays and control structures in assembly language (do-while, if-else, and for loop).

11. Write nested function calls using stack frames and local variables.
12. Write an assembly language program to call recursive functions.
13. Implement C/C++ pointers and references.
14. Implement pass-by-value and pass-by-reference parameter passing.
15. Describe a buffer overflow and why it is a potential security problem.

Grading Criteria:

Category	%
Exams (3)	45
Labs	15
Quizzes	15
ZyBook - Participation Activities & Attendance	15
ZyBooks - Challenge Activities	10

Grade Scale:

%	≥ 92	90-92	88-90	82-88	80-82	78-80	72-78	70-72	68-70	62-68	60-62	< 60
Letter	A	A-	B+	B	B-	C+	C	C-	D+	D	D-	F

Exam Dates:
Exam 1 - Wednesday, October 13
Exam 2 - Wednesday, November 10
Exam 3 - Wednesday, December 15; Friday December 17

Late Policies: Late submissions will not be accepted -- exceptions will be made for medical / personal issues (with a note from a doctor or from the Dean of Students Office).

Absences: It has been scientifically proven that the most significant factor for predicting student success is attendance (although whether this is truly causation or merely correlation is another question). Students should attend each and every class, and notify the instructor ahead of time if you will be absent.

Academic Integrity: The purpose of this course is to teach you in particular about computer architecture and assembly language, and more generally how to think, how to *problem solve*. For that reason, the work that you turn in must be your own. You may have *general* conversations with students to clarify the nature of an assignment, and you can ask for help with debugging, but that second-set-of-eyes-student should not be looking at their code while they assist you. Sometimes students are unaware of whether or not they have committed plagiarism, so here are some tips:

1. if your problem solving begins with ctrl-C and ends with ctrl-V, you have committed plagiarism.
2. if your problem solving starts at Google and ends at Chegg, you have committed plagiarism.

3. if your problem-solving involves surreptitiously glancing at the exam of the student next to you and doing a virtual copy-and-paste, you have committed plagiarism.
4. if that tiny voice inside your head, the same one that inconveniently shuts up entirely while you are working on a computer architecture and assembly language assignment, starts making "ahem" noises, you have committed plagiarism.

Let us consider the pros and cons of committing plagiarism.

Pros:

1. You have completed the assignment.

Cons:

1. You will have missed that exhilarating, ego-boosting, delicious "aha!" moment that everyone experiences when they have, on their own, solved a difficult puzzle.
2. You will be caught, receive a 0 on the assignment/exam, and may face disciplinary action in front of a bevy of grim-faced administrators who you do *not* want to meet.

So what do you do when you can't solve a problem? The answer is simple, ask your instructor for help. We will use the time-tested Socratic method, asking questions that will lead you to the correct answer. Failing that, just write "I have no idea" in bold letters in the comment block at the top of the assignment, turn it in, and you will gain *some* points (more than 0).

For more detailed information on what constitutes academic misconduct, please see the discussion of UWS Chapter 14, Student Academic Disciplinary Procedures.

Workload: The workload for this class is substantial, but so are the rewards -- you will become adept at using Linux, study two programming languages, and be able to impress your friends by your ability to do count to 1024 on your fingers.

Accessibility: Your instructor is committed to ensuring a fair playing field. If you have a disability and need assistance (e.g., a note taker, certain seating, extra time to take tests, adaptive technology, etc.), please register with the Accessibility Center, and we work hard to accommodate your needs.

Non-discrimination and Anti-harassment: Your instructor is committed to maintaining a harassment-free, welcoming classroom, and will not tolerate discrimination on the basis of race, religion, creed, color, sex, gender, identity/expression, ancestry, national origin, age, marital status, relationship to other employees, sexual orientation, disability, veteran's status, membership in the military, arrest/conviction record, political affiliation, or any other protected status.

HATE HAS NO HOME HERE.

Class Participation and Feedback: Your instructor relishes class participation and feedback. If you are lost, please, please ask, during class. You do not need to worry about slowing down the class, and your fellow students, who probably were thinking the same thing but were afraid to ask, will silently or possibly aloud applaud your efforts, as will I.

Masks: All students are required to wear an appropriate mask that covers their mouth and nose when they are in the classroom. They must also adhere to additional expectations communicated by the instructor or posted in the classroom. Note: UWO procedure dictates that, during the COVID-19 pandemic, an instructor cannot begin class until all students are wearing a mask properly. If a student is non-compliant with the masking policy and refuses to leave the classroom promptly when requested, the instructor is required to cancel class. Students responsible for class cancellation for these reasons will be referred to the Dean of

Students office, and the student will be unable to attend class until they meet with the Dean of Students. The student may be dropped from the class by the Dean of Students.

Consumer Information: Students are advised to see the following URL for disclosures about essential consumer protection items required by the Students Right to Know Act of 1990: https://uwosh.edu/financialaid/consumer-information/