

CS 251: Computer Architecture & Assembly Language

Spring 2021

Instructor: Michael P. Rogers
Office: Halsey 214
Office Hours: TuTh 3:00-5:00 PM, Fr 1:30-2:30 PM, or by appointment (all virtual)
Email: rogersm@uwosh.edu
Phone: 920-424-1388 / 309-825-6454

Class Times: MW 9:10-10:10 AM, Halsey 208; F 9:10-10:10 AM, Halsey Lab 101C
Credits: 3

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

Delivery: This class will be taught using the HyFlex model - you may participate face-to-face or online, and transition between the two at your discretion. The class will be synchronous.

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 to learn.zybooks.com
2. Enter zyBook code: UWOSHCOMPSCI251RogersSpring2021
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 (tentative, subject to change):

Exam 1 - Monday, March 8

Exam 2 - Monday, April 12

Exam 3 - Friday, May 14

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: In the same way that you can't get in shape by having your friend run round the block, or experience bliss by watching somebody else complete a crossword puzzle, you cannot learn by copy-and-pasting somebody else's answers.

First, try to solve any problem that you are presented with entirely on your own, without anything other than your text/notes. If you can do that, you truly understand the material. If not (and this definitely will happen unless you're some sort of prodigy), ask your instructor: they wrote the question, they probably know the answer :-).

You may have a *general* conversation (in English, pseudocode, or by drawing pictures) about how to approach a problem, and you may have a peer examine your code to help debug it. You may not look at or use your peer's code, nor turn in code that you find elsewhere: such plagiarism will result in a 0 for the assignment, and other unpleasantness may ensue. See the Dean of Students Office web page [for more information](#).

Exams and quizzes are to be done entirely on your own. These are closed book, closed notes, because this material is so absolutely fundamental that you must have it at your fingertips.

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.

Feedback: Your instructor thrives on feedback. If a concept doesn't make sense, **ask**, and if the answer didn't make sense, **ask again!** There are no stupid questions (and any stupid answers are my responsibility 😊). It may seem intimidating to put yourself out there and admit that you don't understand something, but:

1. if you didn't understand something, there is a good chance that your peers didn't either, and they will (silently, or better yet, by banging on desks like they do in parliament) applaud your act of inquiry
2. you needn't worry about "slowing down the class". Learning never slows down anything.
3. your instructor is intimately familiar with being in situations where things don't make sense: you will receive a very sympathetic hearing

If you do feel uncomfortable asking in class, please ask during office hours.

COVID-19 Policies:

1. Everybody **must** wear a mask/face covering inside all University buildings, practice social distancing, and practice appropriate hygiene, i.e., wash your hands (see, you mother was right :-)
2. This is a HyFlex class: you may attend face-to-face, or online, and transition between the two as you see fit.
3. Classes will be held synchronously: students attending online will connect via Collaborate Ultra during class time.
4. Online students will be expected to take exams during the same time, and in the same fashion, as the in class students.
5. Students who are feeling poorly are asked to follow the advice listed on the [Titans Return](#) page.