CompSci 251 Computer Architecture and Assembly Language Spring'16

Instructor: George Georgiev
Office: HS, Room 217
Phone: 424 - 11 80 **E-mail:** georgiev@uwosh.edu

Office Hours: M W F: 10:20 – 12:30 or by appointment

Section 1:

Lectures: MW: 8:00 am - 9:00 am, NHS 237
Labs: F: 8:00 am – 9:00 am, HS 101 C

Required Text: None

References:
Introduction to Computing Systems: From Bits and Gates to C and Beyond
(Second Edition) by Yale N. Patt and Sanjay J. Patel

Web site for the course: http://www.uwosh.edu/faculty_staff/georgiev/subjects/CSC251/

Current Catalog Description
CS 251 Comp Architecture and Assembly Language (3 units)
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.

Prerequisite: Computer Science 221 with a grade of C or better. (Fall, Spring) Required.

Course Outcomes:
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 high-level language switch statements with jump tables.
14. Implement C/C++ pointers and references.
15. Implement pass-by-value and pass-by-reference parameter passing.
16. Semantics.

Major Topics Covered in the Course

1. Basic computer organization and memory layout, including big-endian versus little-endian byte ordering
2. Integer representation: sign-magnitude, 1's Complement, 2's Complement
3. IEEE 754 Floating Point Standard
4. Concept of an instruction set architecture (ISA)
5. Instruction set of some RISC ISA
6. Assembly language versus machine language
7. The fetch-execute cycle
8. The stack, functions, recursive functions, function calls and register conventions
9. Stack frame and memory for local variables on the stack
10. Pass-by-value and pass-by-reference parameter passing mechanisms
11. Pointers and references in languages like C and C++

If time permits:
12. Objects, classes, and methods of object-oriented programming
13. Memory Hierarchy and Direct-mapped Cache
14. Pipelining

Course Requirements:

There will be three exams, unannounced quizzes, assignments, and laboratory works. The material for all exams will come from either a material covered in class, lab work, and/or programming assignments.

Complete all required work on time. In the event that an exam must be missed, or required work can't be completed on time, due to illness or other serious and unavoidable circumstance, notify the professor as far in advance as possible by phone or e-mail.

The assignments are due by 8 am on the due date (electronic copy (e-mail) is due by 8:00 am, and a paper (hard) copy of the assignment is due at the beginning of the class). Assignments will be accepted up to three days late subject to the following penalties:

Turned in	Penalty
After 9:00 am on the due date	10%
1 day late	25%
2 days late	50%
3 days late	75%

Saturdays, Sundays, and holidays count when computing penalties.

If you work with a partner, you will submit one electronic copy and one paper copy of the assignment with all names on it. The partners will earn equal scores on the assignment. You may work alone on some assignments and with partners on others. You may change partners during the semester.

You are encouraged to discuss assigned problems with other people (teams) but you (team) must design and write own solutions/code for all assignments. Submitting modified versions of other people's (team's) work as own is considered cheating.

There will be no make up for unannounced quizzes.

There will be one make up for the exams, which will cover all topics. It will be at the end of the semester.

Make up exam will be given if you call before the exam, make arrangements, have a medical certificate signed by the physician, and have a note from the Dean of Students Office.

The three exams will be announced at least a week before taking place.

Laboratory assignments will be in the teaching lab. You are encouraged to discuss the lab assignment with others before and during the lab hours, but each student must demonstrate her or his own solution.

Evaluation:

Three Exams:	~60%	(20% each)
Programming assignments:	~25%	(equal points for each assignment)
Unannounced quizzes:	~5%	(equal points for each quiz)
Laboratory Assignments	~10%	(equal points for each lab)

Grading:

Score	Grade
>= 92	A
90-92	A-
88-90	B+
82-88	B
80-82	B-
78-80	C+
72-78	C
70-72	C-
68-70	D+
62-68	D
60-62	D-
< 60	F

Feedback:

Your comments and questions about all aspects of the course (content, grading, teaching methods, pace, textbook, etc.) are welcome. You can use e-mail or talk to me during office hours.