CS 251 - Computer Architecture and Assembly Language - 3 Credits
Spring 2020

Instructor: David Furcy
Email: furcyd@uwosh.edu
Office: Halsey 221
Office Hours: **MWF** 1:40-2:40, **TuTh** 12:30-2:00, or **by appointment**

Class Meetings: **MW [Lecture]** 9:10-10:10 in Halsey 202
F [Lab] 9:10-10:10 in Halsey 101C

Lab Assistant: Zach Novak novakz01@uwosh.edu

Prerequisite: A grade of C or better in CS 221

Class Web Page: canvas.uwosh.edu
You should check this Canvas site daily.

Required Text: We will be using a zyBook for this class. Follow these instructions:

1. Click on a zyBook link in Canvas (e.g., the R1 link in the Day 1 module). Do NOT go to the zyBooks website to create a new account.
2. Subscribe

A subscription is **$77**. Students may begin subscribing on Jan. 20, 2020 and the cutoff to subscribe is May 3, 2020.
Subscriptions will last until May 29, 2020.

Catalog Course 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.*

Since the catalog description above is a bit out of date, here is an **updated course description:**
This course aims to give students an overview of processor and memory hardware, and to teach them how high-level language programs map onto some Reduced Instruction Set Architecture or RISC computer. Students will learn how computer hardware supports the instruction set architecture. Students will be able to analyze why programs behave the way they do and how inefficiencies arise. Students will learn how to implement pointers and references in machine language.

Course Outcomes: At the end 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 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.

Course Grading Policy: Your final grade for this course will be based on the following seven components:

a) **Daily reading assignments (R1, R2, etc.):** You will be assigned a selection of sections from our zyBook to read before 8:30AM on each class day (MWF). Each reading assignment will include participation activities (PAs) that must also be completed by the same deadline as the reading. Your daily scores on these activities will be added up to make up your final score for this component. On some days (announced in advance) the reading assignment and associated PAs will only be STRONGLY recommended but not required (and thus not graded). This will most likely happen on some Fridays (i.e., lab days) when the reading assignments may be much longer (but mostly review material), e.g., one or two full chapters in the zyBook.

b) **Daily challenge activities (C1, C2, etc.):** On most days, you will be assigned a selection of challenge activities (CAs) from our zyBook, namely those that appear in the sections that were part of the previous reading assignment. For example, you will be required to read the sections in R5 (and to complete the associated PAs) before 8:30AM on Day 5 of the semester (that is, before we discuss these sections in class on Day 5) and then to complete the CAs (called C5) contained in R5 before 8:30AM on Day 6, that is, on the next class day. You are given two or three extra days to complete the CAs to make sure that you have time to reread both the zyBook sections and your class notes and to ask lingering questions before being testing on the material covered by the CAs. Your daily scores on these CAs will be added up to make up your final score for this component.

c) **Weekly labs (L1, L2, etc.):** Each Friday, we will meet in HS 101C for hands-on, programming activities in C or MIPS on Linux. In addition to the reading assignment for that day (always due at 8:30AM before the lab), there will be a lab assignment to be completed and submitted to Canvas. Lab assignments are intended to be completed during the lab session and are officially due at 1:40PM on Friday. However, each week, you will be granted a 100-hour penalty-free extension for each lab until 5:40PM of the following Tuesday. Therefore, this will be the unofficial (but hard) deadline for each lab assignment. However, note that reading assignments due on a lab day are still due BEFORE the lab session, as stated above, since such reading assignments are meant to be pre-requisites to the lab activities. Unlike for PAs and CAs, all lab assignments will be

given equal weight. Your weekly scores on these labs will be averaged to make up your final score for this component.

d) **Weekly quizzes (Q1, Q2, etc.):** You will have to complete a weekly quiz on the material covered during the week(s) prior to quiz day. These quizzes will be completed in exam conditions (with no notes or calculator allowed) at the end of a regular class session. All quizzes will be equally weighted and your weekly scores will be averaged to make up your final score for this component.
e) **Three exams (E1, E2, E3):** There will be three one-hour exams taken during regular class periods. The final exam will take place during week 14. The first two exams will take place around weeks 6 and 10, respectively. Each exam date will be announced in class and on Canvas at least one week before the exam.

Your overall numerical grade for the course will be computed as the weighted sum of the component grades using the following weights:

Component	Weight
Reading assignments (and associated PAs)	15%
Challenge activities	10%
Labs	15%
Quizzes	15%
Exam #1	15%
Exam #2	15%
Exam #3	15%

Your final letter grade for the course will be computed using the following mapping:

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

While this overall grading scheme is fixed, I will be happy to discuss any issue you may have with individual grades. If you notice a mistake or have a question regarding a specific grade, please contact me *as soon as possible*. Do not wait until the end of the semester to bring up grading issues.

Attendance and participation: You are expected to not only attend **every** class meeting but also to come **prepared** for and participate actively in it. Necessary preparation requires you to have studied and assimilated the material covered in previous sessions, to have met with the instructor outside of class to discuss any questions you may have, to have completed the reading assignments and associated zyBook activities, and to have completed prior lab assignments on time.

Note that the reading assignments and associated zyBook activities, i.e., PAs (resp., CAs), to be completed before (resp., after) each class, together account for 25% of your overall course grade.

In fact, they are even more important to your success since the learning they foster will more or less directly impact your performance in the course as a whole, which will be tested in quizzes and exams.

It is hard to imagine how a student could do well in this course while missing classes or assignments, attending classes unprepared, or not participating during/between them.

On the positive side, I have high expectations for my students and will always support and encourage you. I **strongly encourage** you to **ask any question** or raise any issue you have with the course either during or at the end of class, or during my office hours. I will also gladly meet with you by appointment. Send me email to make an appointment. While I will meet with you as soon as my schedule permits, do not expect me to be widely available just before a big deadline, in case many students ask for help at the last minute.

Late submissions: THERE WILL BE NO LATE SUBMISSIONS ACCEPTED ON ANY COMPONENTS OF THIS CLASS. If your submission reaches me after the due date/time (even if it is late by only a few seconds, as recorded by Canvas), I will not grade it and you will receive a **zero**. Late submissions can easily be avoided by starting to work on each assignment right away and asking questions early if you get stuck.

The zero-score for late submissions can be waived in **only one** scenario, namely if you give me a signed note from a doctor or a written justification for the extension from the Dean of Students Office. If you miss an exam, you **may** be able to take a make-up exam provided you give me a valid justification (see above) ahead of time if possible. Only one make-up exam will be given. It will be a comprehensive exam scheduled at the end of the semester. If you miss a quiz, you **may** be able to take a make-up quiz, provided you have a valid justification for your absence.

Collaboration versus Cheating: All submissions, unless otherwise stated, must be the work of only one student, namely the one whose name appears on the submission. While it is acceptable and encouraged to discuss the assignments with others, you must submit your own work (or that of your team only, when applicable) unless you can live with a zero and the other potential academic sanctions of cheating. Check out the UWO Student Academic Disciplinary Procedures (UWS 14) at https://www.uwosh.edu/stuaff/images/Chapter%20UWS%2014.pdf/view for details.

Accommodations: The University of Wisconsin Oshkosh supports the right of all enrolled students to a full and equal educational opportunity. It is the University's policy to provide reasonable accommodations to students who have documented disabilities that may affect their ability to participate in course activities or to meet course requirements.

Students are expected to inform instructors of the need for accommodations as soon as possible by presenting an Accommodation Plan from either the Accessibility Center, Project Success, or both. Reasonable accommodations for students with disabilities is a shared instructor and student responsibility.

The Accessibility Center is part of the Dean of Students Office and is located in 125 Dempsey Hall. For more information, email accessibilitycenter@uwosh.edu, call 920-424-3100, or visit the Accessibility Center Website.

Disclosure: 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/

In conclusion, remember that computer science classes require a lot of work in addition to active participation in class. It takes considerable practice to develop the technical and analytical skills targeted by this course. You will need to spend **at least, and typically much more than, three hours of effort outside of class for each in-class hour**. Having said this, I expect every hardworking student to do well in this course.

Have fun this semester and good luck!