Oregon State University
Ecampus

Course Name: Computer Architecture & Assembly Language
Course Number: CS 271
Credits: 4
Instructor Info: Stephen Redfield / Stephen.Redfield@oregonstate.edu
Instructor Info: Eric Vogel / Eric.Vogel@oregonstate.edu

Teaching Assistants' names and contact info:

- TAs and Office Hours Information on Course Syllabus Page (in Canvas)

Textbook: None

Course Description

Introduction to functional organization and operation of digital computers. Coverage of assembly language; addressing, stacks, argument passing, arithmetic operations, decisions, macros, modularization, linkers and debuggers.

Prerequisites

Required: CS 151, 161 or 165
Recommended: CS 162 & CS 225

Communication

1. Please post all course-related questions in the Piazza page so that the whole class may benefit from your conversation. Not all posts require a reply from the instructor/TA and often it is better for students to hash out an answer to a question.
2. To directly contact the TAs open Canvas Inbox, compose a new message, select this course, then next to the 'To' box click the address book, select "Teaching Assistants", then choose the name(s) of TA(s) you wish to contact.
3. Please email your instructor only for matters of a personal or private (grading) nature. The instructor or a TA will reply to most course-related questions within 24-48 hours.
4. Any email sent to the instructor about this course **must** originate with an OSU supplied email account *and contain the tag* **[CS271]** at the beginning of the subject. Failure to comply with this will result in delayed (or possibly nonexistent) response to your email.
5. Office hours will be held online through the class Slack workspace. To get started on Slack please go to https://it.engineering.oregonstate.edu/slack. You'll want to add the class-specific Slack by connecting to the **oregonstate.enterprise.slack.com** workspace. It should then show up as one of your connected Workspaces.

Canvas & Piazza

1. This course will be delivered via Canvas and you will interact with your classmates and with your instructor through Piazza. Within the course Canvas site you will access the learning materials, such as the syllabus, assignments, projects, and quizzes. Class discussions will be on Piazza.
2. To preview how an online course works, visit the Ecampus Course Demo. For technical assistance, please visit Ecampus Technical Help.
3. Canvas is optimized for the most recent versions of most popular browsers. If your browser of choice is an out-of-date version, you should update for use with Canvas, especially for quizzes and exams. If you are having browser troubles, seek out the Technical Assistance described above.
4. If your device has trouble dealing with Canvas (as some tablets do), make sure you have an alternative available for things like quizzes and exams. If you are having device troubles, seek out the Technical Assistance described above. Telling me the day after an exam has ended that you had browser issues on your smartphone/tablet is unlikely to get you what you want.
5. General announcements for the class, will be sent as emails through Canvas and announcements on Piazza. It is your responsibility to keep up with messages in Canvas Inbox, and in Piazza. You should check the course Piazza at least a couple times per week, just in case.

Technical Assistance

If you experience any errors or problems while in your online course, contact 24-7 Canvas Support through the Help link within Canvas. If you experience computer difficulties, need help downloading a browser or plug-in, or need assistance logging into a course, contact the IS Service Desk for assistance. You can call (541) 737-8787 or visit the IS Service Desk online.

Computing Requisites

Computers used for this class will need to run a Microsoft Windows environment on a x86 or x86-64 processor. Students with Mac computers which use x86-type processors may use a virtual desktop or dual-boot setup to install Windows. Students with Mac computers on other architectures will need to use a remote desktop environment. See the course Tools page (via the Syllabus page) for details.

Measurable Student Learning Outcomes

1. **Access** and **Interpret** binary data stored in memory
2. **Illustrate** the Instruction Execution Cycle
3. **Create** and **Analyze** well-modularized assembly language programs utilizing decision, repetition, and procedure structures
4. **Utilize** a debugger to Identify and Correct bugs in assembly language programs
5. **Illustrate** the System Stack as it is used for procedure calls and parameter passing
6. **Illustrate** the primary components of a modern Computer Architecture, and **Explain** their function.

Evaluation of Student Performance

Syllabus Quiz	1%	Open Resource, ∞ attempts, unlocks first module
Project #0	2%	VS Setup Assignment / Proof
Project #1	6%	Your own work only
Project #2	3%	Debugging Lab, may brainstorm together
Project #3	6%	Your own work only
Project #4	6%	Your own work only
Project #5	8%	Your own work only
Project #6	10%	Your own work only
Module Summary Exercises (x10)	20% (2% each)	Open Resource, 2 attempts, 24 hour limit
Quizzes (x2)	10% (5% each)	Open Notes, 1 attempt, 1 hour limit
Midterm	12%	**PROCTORED**, 1 hr 50 min limit, notesheet only
Final	16%	**PROCTORED**, 1 hr 50 min limit, notesheet only

Letter Grade Scale:

Percentage	**Grade**
93+	A
90 – 92.99	A-
87 – 89.99	B+
83 – 86.99	B
80 – 82.99	B-
77 – 79.99	C+
73 – 76.99	C
70 – 72.99	C-
67 – 69.99	D+
63 – 66.99	D
60 – 62.99	D-
0 – 59.99	F

- If you end up with a grade average of 89.99%, you will get a B+ in the class.

- The intention is to give you a lot of practice before the exams. This practice helps you perform better and more quickly on the exams and programming projects.
- I know taking this many assessments is not your favorite thing to do, but I have found that frequent assessments help students keep pace with the work and provide quick feedback on how well the material is understood or remembered.

Programming Projects

- The programming projects are a significant portion of this class, and are also the most common place for students to struggle. Several things about this class' programming projects may be new to you.
    - You may not have used Visual Studio before.
    - Intel x86 Assembler code will be new, and programming at the assembler level is very different from using higher level languages.
    - Stepping through the assembler code in the debugger will be new.
- Don't be lulled into over-confidence from easy early assignments only to be surprised by later assignments. Watching the clock tick past midnight for an assignment that feels far from working is not enjoyable. It causes stress, and stress is bad, mmkay?
- All source files (.asm files) must include a comment block at the top that contains your name and OSU email address, the class number/section, the assignment name, due date, and program description in your own words. Neglecting this information is a significant point loss on any assignment. See the CS271 Style Guide for more.
- You can overcome a poor grade on one assignment and still do well in the class. Do not allow struggling on one programming assignment to cause you to be late on all programming assignments.
- Don't miss submitting a programming assignment. You are much better off to submit a partially functional assignment than to not submit anything for an assignment.
- You must submit all your assignments through Canvas.
    - Submit your work for each assignment as a **single asm** file through Canvas. You should not need to submit any additional files for a programming assignment. If you use external library other than the Irvine32 library your code will fail to assemble and link in the standard Visual Studio environment that we use to grade the assignment. That means you will be disappointed with your grade. We expect you to make use of the Irvine32 library and no other libraries in your code. If you need to comment on your code, place your comments into the asm file.
    - You can submit your assignments more than once through Canvas. Each will be time stamped. We will grade only the last one submitted.
- **We reserve the right to ask you to explain a complicated piece of code**. If you cannot explain your own code to us, you may be investigated for violation of student conduct.

- If you are struggling on a programming assignment, the first thing you should do is make sure you have worked through the assigned and relevant explorations, and reviewed the notes you've taken.
    - Sending your entire source code to the instructor or TA with a note saying "Something wrong can you fix it?" is unlikely going to get you the response you want. The instructors and TAs are not debuggers.
    - Run the program in the Visual Studio debugger yourself. The way to get better at debugging code is to use the debugger.
    - Make sure you read the entire assignment. There is useful information in all that text.
- We will be using Visual Studio as the development environment for this class, using MASM (Microsoft Macro Assembler). If you use some other assembler (e.g. NASM), your code likely will not assemble and you'll lose most/all of your points. If you don't already have Visual Studio, you can get it free for student use. Check the Tools tab of the Canvas Syllabus Page.
- Your programming assignments must run in Visual Studio to be graded. If your assignment does not run in Visual Studio, then you will get a zero for a grade. Running under some other assembler or emulator in addition to VS is fine, but it must still run in Visual Studio.
- **Feedback** for your programming assignments will be given through Canvas Rubrics. This is important feedback. You don't want to repeat any errors on following assignments. The rubric will also identify by whom your assignment was graded, making it much easier to contact her/him if you have questions about your grade.
- If you are unable to locate the feedback on your assignments, ask a TA to guide you to it.
- If you wish to petition a grade, you must do so within one week of receipt of the grade, by email to your instructor.

Module Summary Exercises:

- Module summaries are open book, open note, open Internet, and open lecture. You can use just about anything, including your fellow students, while taking a module summary. The module summary exercises are not proctored.
- Module summaries will primarily (not exclusively) cover assigned explorations from their module.
- Module summaries have a time limit of 24 hours.
- You will be able to take each module summary at most twice. The recorded score will be the higher of the two scores you receive.
- Module summaries <u>cannot</u> be taken after the due date.
- The Module summaries are intended to help you pull together the material from the week. This will help you with the quizzes, the exams, and (importantly) the programming projects.

Quizzes:

The quizzes are intended to help you pull together the material from the previous couple of weeks, and to give you some live practice for subjects that will be on the midterm and final.

- Quizzes are open note, open Internet, and open exploration. You can use just about anything **except** your fellow students while taking a quiz. You will be able to **take each quiz once**. They are not proctored and cannot be taken after the due date.
- Quizzes become available on **Thursday** of the week they are due.
- **Quizzes are timed**. You won't be able to exceed the time limit on the quizzes. It is not the intent of the quizzes to be time pressured, but pace yourself. The rate at which you are able to complete the quizzes will give you a good measure on how quickly you can move through the midterm and final.

Exams (Midterm and Final)

- The only resources allowed on exams are a blank scratch paper, a single sheet (8.5x11) of handwritten or typed notes (double sided), any handheld calculator (no phone calculators) and a specific web-based calculator, the link for which will be available in the course materials and in the exam itself.
- Exams have windows of availability of 4 days to allow some flexibility in scheduling.
- Most students use the entire time available (110 min) for the exams. The better you prepare for the midterm and final, the easier it will be for you to complete them within the allotted time.
- The Midterm and Final Exam for this course are Proctored exams via Proctorio. Ensure you take the Proctorio Practice Exam at least a week (but better 2+ weeks) in advance to guarantee everything is working correctly.
- Please show your notes/scratch paper to the camera before and after the exam.

Using Proctorio Automated Proctoring for Exams

- This course will use an automated online proctoring system called Proctorio, where your exam session is recorded for instructor review. You will not need to schedule proctoring appointments, and there is no cost to you to use Proctorio.
- Please note that a functioning webcam and microphone are required for using Proctorio. If you do not have these, you will need to locate and submit an alternative proctor through the exams and proctoring form and pay for any associated proctoring fees.
- Your security and privacy are important. You can read more about Proctorio's privacy and data security policies on their website, and more information about using this tool can be found in the course site

Late Work Policy

***2021 NOTE:** The normal late work policies are expanded due to the extreme circumstances of, well, *everything* right now. The total number of grace days has been expanded from two (2) to five (5). In addition, if you are adversely impacted by the pandemic and it prevents you from completing work on time, please email your instructor to get in touch with us!

- Summary Exercises, Quizzes, and Exams **may not** be submitted late.
- **Programming Assignments** have exactly two (2) days from the due date, no more, to be submitted. Since programming assignments are normally due on a Sunday, 2 days late makes that Tuesday. Late work is penalized 15% (of the earned points) per day. Any programming project submitted more than 2 days after the due

date will receive a grade of zero (0). Don't make the mistake of submitting your assignment late just trying and get the last few points by making it perfect. *Perfection is the enemy of done.*

- You may make use of **five*** grace days for submission of programming projects, used in increments of **one day**. The grace days remove the 15%/30% penalty but do not further extend the submission deadline. You must still submit the assignment **at most** 2 days late. Grace days **may*** be used on the final program.
- Grace days **must** be invoked *before the assignment is due*, and **must** be invoked via the following process. Failure to follow this procedure renders the grace days expended but invalid.
    - Before the assignment is due, in Canvas, go to "Grades".
    - Select the assignment (e.g. Program 1).
    - On the right side of the screen, in the "Add a Comment" box enter
    - "X Grace Days" where X is the number of grace days you will use for that assignment.
- If you comment that you plan to use X grace days (e.g. "2 Grace Days") but find you only need X-Y grace days, make another assignment comment stating "Revoke Y Grace Days" (e.g. "Revoke 1 Grace Day") to recover the unused grace day. You may not re-apply grace days for the same assignment after revoking them.
- I encourage you to not use up your grace days early in the term. The programming assignments get harder as the as the term progresses. Start your programming assignments as soon as possible. Do not wait until the last weekend to begin them.

Makeup Exams

Makeup exams will be given only for missed exams excused **well in advance** by the instructor. Excused absences will not be given for airline reservations, routine illness (colds, flu, stomach aches), or other common ailments. Excused absences will generally not be given after the absence has occurred, except under very unusual circumstances.

Incompletes

Incomplete (I) grades will be granted only in emergency cases (usually only for a death in the family, major illness or injury, or birth of your child), and if the student has turned in 80% of the points possible (in other words, usually everything but the final paper). If you are having any difficulty that might prevent you completing the coursework, please don't wait until the end of the term; let me know right away.

Academic Honesty

- Students are expected to do their own work. The only sources you're allowed to use code from are the explorations, and you must make a comment with the Module/Exploration. Direct use of any other resources is prohibited.
- Programming assignments present unique challenges for graders. It is often difficult for a grader to distinguish between legitimate help and plagiarism. We use plagiarism-detection software to check your code against the code from other students. It is quite sophisticated and can see through variable name changes and formatting differences
- Honesty is essential for learning to take place. It will form the foundation of your professional integrity in your career.

- Specific Examples:
  - You **may** ask conceptual questions related to optimizing your code on Slack or Piazza.
  - You **may** post Module Summary questions and solutions at any time to Slack or Piazza.
  - You **may** post Quiz questions and solutions ***after the due date*** to Slack or Piazza.
  - You are **encouraged to** discuss course content with other students, TAs, the instructor, or anyone else who will listen, including general discussion of homework assignments and how to fix specific issues.
  - You **may** share pseudo code and ideas about how to solve or approach problems. If you are getting odd assembler messages, you can share the snippet of code that is producing the message; you don't need to share the entire file.
  - You **may not** copy anyone's (or allow someone to copy your) solutions. It is possible to discuss problems without plagiarizing. One of the best methods of debugging is to explain your solution to someone else.
  - You **may not** post any complete functions/procedures/logic blocks to Slack or Piazza.
  - You **may not** post any exam questions or solutions in any form.
  - You **must** make any git repo you post this code on private, with the exception of the Portfolio assignment.
- If you are found in violation of any of the above policies, whether you are the giver or receiver of help, you will be subject to the University Academic Misconduct process. The first offense usually results in a warning and an assignment penalty (0-grade); the second offense can result in a disciplinary hearing, and possibly removal from the Program.
- The bottom line is: <u>Each student is expected to understand all aspects of the programs they submit for credit</u>.
- The following are examples of plagiarism, drawn from actual submitted and penalized cases.
  - Student Googled the problem and ended up finding a code written online. Student used the code there to improve their own work.
  - Student worked with a classmate, each tackling one half of the program. Both students modified the identifiers, labels, comments, etc...
  - Student found a solution to the problem online and changed a few variable names and methods and handed it as their own.

Academic Integrity

- The Code of Student Conduct prohibits Academic Misconduct and defines it as:
  - Any action that misrepresents a student or group's work, knowledge, or achievement, provides a potential or actual inequitable advantage, or compromises the integrity of the educational process.

- To support understanding of what can be included in this definition, the Code further classifies and describes examples of Academic Misconduct, including cheating, plagiarism, assisting and others. See the Code of Student Conduct for details.
- You are expected to do your own work and demonstrate academic integrity in every aspect of this course. Familiarize yourself with the standards set forth in the **OSU Code of Student Conduct Section 4.2.** You must only access sources and resources authorized by the instructor. You may not show your work to any other current or future students without the instructor's authorization. Violations of these expectations or the Code of Student Conduct will be reported to the Office of Student Conduct and Community Standards. If there is any question about whether an act constitutes academic misconduct, it is your responsibility to seek clarification and approval from the instructor prior to acting.

Statement Regarding Religious Accommodation

Oregon State University is required to provide reasonable accommodations for employee and student sincerely held religious beliefs. It is incumbent on the student making the request to make the faculty member aware of the request as soon as possible prior to the need for the accommodation. See the Religious Accommodation Process for Students.

Guidelines for a Productive and Effective Online Classroom

(Adapted from Dr. Susan Shaw, Oregon State University)

Students are expected to conduct themselves in the course (e.g., on discussion boards, email) in compliance with the university's regulations regarding civility. Civility is an essential ingredient for academic discourse. All communications for this course should be conducted constructively, civilly, and respectfully. Differences in beliefs, opinions, and approaches are to be expected. In all you say and do for this course, be professional. Please bring any communications you believe to be in violation of this class policy to the attention of your instructor.

Active interaction with peers and your instructor is essential to success in this online course, paying particular attention to the following:

- Unless indicated otherwise, please complete the readings and view other instructional materials for each week before participating in the discussion board.
- Read your posts carefully before submitting them.
- Be respectful of others and their opinions, valuing diversity in backgrounds, abilities, and experiences.
- Challenging the ideas held by others is an integral aspect of critical thinking and the academic process. Please word your responses carefully, and recognize that others are expected to challenge your ideas. A positive atmosphere of healthy debate is encouraged.

Expectations for Student Conduct

Student conduct is governed by the university's policies, as explained in the Student Conduct Code (https://beav.es/codeofconduct). Students are expected to conduct themselves in the course (e.g., on discussion boards, email postings) in compliance with the university's regulations regarding civility.

Personal Issues Conflicting with Course Progression

Your advisors are excellent resources when significant personal issues conflict with your ability to progress in the course. Remember to keep them (and us) updated in these situations.

Statement Regarding Students with Disabilities

Accommodations for students with disabilities are determined and approved by Disability Access Services (DAS). If you, as a student, believe you are eligible for accommodations but have not obtained approval, please contact DAS immediately at 541-737-4098 or at http://ds.oregonstate.edu. DAS notifies students and faculty members of approved academic accommodations and coordinates implementation of those accommodations. While not required, students and faculty members are encouraged to discuss details of the implementation of individual accommodations.

Course Content

Week	Topics	Week	Topics
1	• Introduction to Assembly Language • Internal Data Representation • Binary/Hexadecimal Arithmetic • Computer Architecture Fundamentals • IA-32 Basic Execution Environment	6	• Error Detecting Codes
2	• MASM Assembly ○ Introduction ○ How Instructions Work ○ Defining Data (Variables) ○ Arithmetic Operations • Irvine Procedures	7	• MASM Assembly ○ Parameter Passing (Stack) ○ Addressing Modes ○ Arrays & Array Parameters
3	• MASM Assembly ○ Conditions and Control Structures ○ Repetition Structures • Using the Visual Studio Debugger	8	• MASM Assembly ○ Data-related operators ○ String Processing ○ Macros
4	• MASM Assembly ○ Defining Constants ○ Data Validation • Endianness • IEEE 754 Floating Point Format	9	• Reverse Polish Notation (RPN) • IA-32 floating-point unit (FPU)
5	• The Runtime Stack • Program Design (Modularization) • MASM Assembly ○ Procedure Calls/Returns ○ Procedure Documentation ○ Passing Parameters to Procedures	10	• CISC vs. RISC Architectures • Parallelism

Accessibility of Course Materials

All materials used in this course are accessible. If you require accommodations please contact Disability Access Services (DAS).

Additionally, Canvas, the learning management system through which this course is offered, provides a vendor statement certifying how the platform is accessible to students with disabilities.

Tutoring and Writing Assistance

The OSU College of Engineering has in-College tutors available for Computer Science students. To sign up for this service, please visit the OSU CoE Academic Support site.

The Oregon State Online Writing Suite is also available for students enrolled in Ecampus courses.

Ecampus Reach Out for Success

University students encounter setbacks from time to time. If you encounter difficulties and need assistance, it’s important to reach out. Consider discussing the situation with an instructor or academic advisor. Learn about resources that assist with wellness and academic success.

Ecampus students are always encouraged to discuss issues that impact your academic success with the Ecampus Success Team. Email ecampus.success@oregonstate.edu to identify strategies and resources that can support you in your educational goals.

- **For mental health:**

  Learn about counseling and psychological resources for Ecampus students. If you are in immediate crisis, please contact the Crisis Text Line by texting OREGON to 741-741 or call the National Suicide Prevention Lifeline at 1-800-273-TALK (8255).

- **For financial hardship:**

  Any student whose academic performance is impacted due to financial stress or the inability to afford groceries, housing, and other necessities for any reason is urged to contact the Director of Care for support (541-737-8748).

Student Evaluation of Courses

During Fall, Winter, and Spring term, the online Student Evaluation of Teaching system opens to students the Wednesday of week 8 and closes the Sunday before Finals Week. Students will receive notification, instructions and the link through their ONID email. They may also log into the system via Online Services. Course evaluation results are extremely important and used to help improve courses and the learning experience of future students. Responses are anonymous (unless a student chooses to “sign” their comments, agreeing to relinquish anonymity) and unavailable to instructors until after grades have been posted. The results of scaled questions and signed comments go to both the instructor and their unit head/supervisor. Anonymous (unsigned) comments go to the instructor only.