CSE 31
Computer Organization

Lecture 1 – Course Info

Who Are We?

- Instructor
  - Santosh Chandrasekhar
  - schandrasekhar@ucmerced.edu
  - Office Hours (in-person and via Zoom):
    - » **Posted in syllabus**
    - » By appointment

- TAs
  - Akshay Bhatia: abhatia8@ucmerced.edu
  - Zhixun He: zhe5@ucmerced.edu

Email Policies

- All email inquiries received before 5pm during school days will be replied within 48 hours

- I may not be able to answer emails received after 5:00pm on weekdays, or during weekend/holidays. Please plan accordingly

- IMPORTANT!
  - You **must** follow the guidelines listed in the email etiquette document available in CatCourses for ALL email communications in this course

Course Guidelines

- Your success is my success
  - This course is only successful when you understand and master the material being presented

- Please ask questions
  - Raise your hand, or just speak up when prompted
  - Don't be shy, you are not being graded for raising a point or asking for clarifications

- Please bring your comments, suggestions, interesting points, or even disagreements to my attention during meetings (office hours, after lecture or by appointment)
  - I/We can learn from you too
  - New ideas and discussions make the class livelier and more interesting

Course Overview

- CatCourses
  - Check regularly for announcements.
  - Labs, Projects, Reading and Homework Assignments will be posted and submitted there.
  - Grades for assignments will also be found there (secure).
- 2 Lectures and 1 Lab per week
- 2 Midterm exams (Week 8 and 13, tentative)
- Final exam (Wed, 10-MAY)
- 10 lab assignments
- 6 Homework and 8 Reading assignments
- 2 programming projects

Grading

- Participation: 15%
  - Lab Participation + zyBook Participation Activities
- Homework: 20%
  - zyBook Challenge Activities
- Lab assignments: 15%
- Projects: 15%
- Mid-terms: 20% (10% each)
- Final exam (comprehensive): 15%

- Grading scheme (tentative):
  - ≥ 87% scores at least an A-
  - ≥ 76% scores at least a B-
  - ≥ 65% scores at least a C-

Course Material

- Text Books:
    - Computer Organization and Design by ZyBooks
        - » Sign up at zybooks.com using your UC Merced Email ID
        - » Enter zyBook code: **UCMERCEDCSE031ChandrasekharSpring2023** (case sensitive)
        - » Subscribe
        - » **You must subscribe to your own copy with your UCM email.**
            - Participation and Homework grade will be evaluated based on the activities within the subscription account.
    - The C Programming Language, Kernighan and Ritchie (K&R), 2nd edition
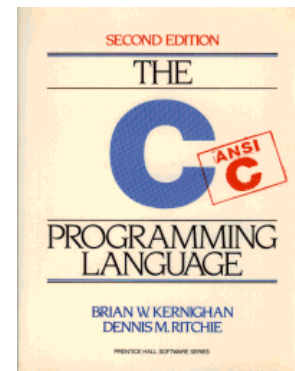
Prerequisites

- CSE30: Data Structures

- Math: logarithms, series, boolean logic, matrices, calculus …

- Coding: intermediate programming experience (Java, C, C++, …)
  - Coding in terminals??

- Curiosity: observe how the world is run by computers, and what problems we face.

Course Objectives

- Learning C
  - If you know one, you should be able to learn another programming language largely on your own
  - If you know C++ or Java, it should be easy to pick up their ancestor, C
- Assembly Language Programming
  - This is a skill you will pick up, as a side effect of understanding the Big Ideas of computer organization
- Hardware design
  - We'll learn just the basics of hardware design

Announcements

- Today: Introductory Material, Course Details
- Lab
  - Lab 1 (Introduction to C) assigned next week (1/22 – 1/28)
    » Due in one week (with one week grace period)
    » Make sure to demo your work to YOUR OWN TA (or me) after submission
    » Demo is REQUIRED to receive credit for assignment (more details later)
- Reading assignment
  - Chapter 4-6 of K&R (C book) to review C/C++ programming

Labs

- Attendance is mandatory
  - Participation grade is evaluated based on lab attendance. Tentatively, **> ~80% attendance or** 11 sessions to receive full lab participation grade (with exceptions). See details on CatCourses (assignment called Lab Participation) posted by the end of this week.
  - Your TA will record attendance for each lab session
- Submit on CatCourses before the assignment goes offline
  - Typically grace period of 7 days after due date
  - Submission made during grace period will incur 10% penalty
  - No submission after grace period (exceptions may apply if approved by instructor beforehand).
- To ensure that your assignments receive a grade, you MUST show/demo your work to your TA or instructor before submission period ends, and we will ask you questions related to your work.
  - Demo MUST be done AFTER submission but BEFORE end of grace period.
  - Submissions without a demonstration will receive a grade of ZERO for individual coding portion of assignment

Labs (cont.)

- Projects
  - 1 – 2 students per group.
  - All group members must submit their own solution in their CatCources account.
  - Same submission policy as labs, except for later deadline and longer grace period
- Try to debug yourself before asking questions
- Follow the guidelines (see below) to debug and ask your TA for help.
  - http://www.cplusplus.com/forum/articles/28767/
  - http://www.catb.org/esr/faqs/smart-questions.html

Code Demo

- **Be prepared**
  - May be asked to perform a walkthrough of code that involves presenting the program in a step-by-step manner to the TA or instructor and answering any assignment-related questions that are posed
  - Questions about lab assignments or projects can be wide-ranging. Eg.,
    - » Explain portions of code in detail,
    - » Provide reasons behind decisions and choices,
    - » Predict program behavior when modifications are introduced, etc.
  - Questions will be used to test knowledge of programming and theoretical concepts relevant to lab assignment or project being evaluated

Code Demo

- You can demo your code only to your OWN TA during the following times:
  - Lab hours (found in [UC Merced Schedule](#) or Syllabus)
  - Office hours (posted in Syllabus by end of this week)
  - By appointment
- If no TA is available (or on direction by a TA), you may demo your code during instructor's office hours

Policies

- Don't copy someone else's code
- Don't give your code away
- Don't outsource your assignments
- No Cheating!
  - Communicating information to another student during examination.
  - Knowingly allowing another student to copy one's work.
  - Offering another person's work as one's own.
  - Penalties are harsh!

Academic Integrity

- Academic integrity is the foundation of an academic community. Academic integrity applies to research as well as undergraduate and graduate coursework.

- Academic misconduct includes, but is not limited to cheating, fabrication, plagiarism, altering graded examinations for additional credit, having another person take an examination for you, or facilitating academic dishonesty or as further specified in this policy or other campus regulations.

- For more information, please visit http://osrr.ucmerced.edu/ and review the following documents
    - » UCM Code of Student Conduct
    - » UCM Academic Honesty Policy

Academic Integrity

- **Cheating** is the unauthorized use of information in any academic exercise, or other attempts to obtain credit for work or a more positive academic evaluation of work through deception or dishonesty.

- **Plagiarism** refers to the use of another's ideas or words without proper attribution, or credit.

- **Collusion** is when any student knowingly or intentionally helps another student to perform any of the above acts of cheating or plagiarism.

  - Students who collude are subject to discipline for academic dishonesty.
  - No distinction is made between those who cheat or plagiarize and those who willingly facilitate cheating or plagiarism

Academic Integrity

- Please note that with respect to the programming assignments we run your code through a system to detect similarity with other projects submitted by your classmates, online solutions, and a database of previous year's submissions.

- The algorithm analyses the structure and flow of the code, so simply changing the variable names and introducing minor changes will not be effective to defeat it.

- You would have to modify the code so much, that you are better off writing the code on your own.

- So **caveat emptor**.

Academic Integrity

- **Cheating vs. Collaboration**: So how do you draw the line between collaboration and cheating? Here's a reasonable set of ground rules. Failure to understand and follow these rules will constitute cheating and will be dealt with as per university guidelines.
  - **The Simpson's Rule**: This rule says that you are free to meet with fellow student(s) and discuss assignments with them. Writing on a board or shared piece of paper is acceptable during the meeting; however, you should not take any written (electronic or otherwise) record away from the meeting. This applies when the assignment is supposed to be an individual effort or whenever two teams discuss common problems they are each encountering (inter-group collaboration). After the meeting, engage in a half hour of mind-numbing activity (like watching an episode of the Simpsons), before starting to work on the assignment. This will assure that you are able to reconstruct what you learned from the meeting, by yourself, using your own brain.
  - **The Freedom of Information Rule**: To assure that all collaboration is on the level, you must always write the name(s) of your collaborators on your assignment.

Computer Science Department Policy on Academic Honesty

- Effective Fall 2019: how Academic Honesty Policy (AHP) will be implemented when students enrolled in classes offered by CSE

- For more details, see [Computer Science Department Policy On Academic Honest - Fall 2019](#)

- First Infraction
  - Student receives a 0 (or equivalent grade) for the assignment
  - Reported to the Dean of the School of Engineering
  - Reported to Office of Student Conduct for review of possible violations of the Code of Student Conduct

- Additional Infractions
  - If one or more prior violations of the AHP (in the same or another course), student receives a failing grade (F) for the course WITHOUT THE ABILITY TO WITHDRAW FROM CLASS

Hints for success

- Attend lecture

- Read the textbook and work on the activities

- Do & understand the assignments YOURSELF

- Create a portfolio to save all your work

- Take notes while reading and in lecture

- Ask questions: We are here to help you!

- Helpful resources posted on CatCourses

- Enjoy Learning!

Welcome to the Class!