Spring 2-1-2018

CSCI 361.01: Computer Architecture

Jesse Johnson
University of Montana, Missoula, jesse.johnson@umontana.edu

Computer Architectures
CSCI 361
Spring 2018 Syllabus

> We can only see a short distance ahead, but we can see plenty there that needs to be done.
>
> –Alan Turning

Instructor Details

Name:	Jesse Johnson
Office:	406A Interdisciplinary Science Building
Telephone:	(406) 243-2356
Email:	jesse.johnson@umontana.edu
Web:	http://hs.umt.edu/hs/faculty-list/faculty-details.php?id=540
Office Hours:	MW 15:00–16:00 , Interdisciplinary Science Building 406A *Or, by appointment.*

Prerequisites

Students taking this course are expected to have:

- Programming experience demonstrated by passing CSCI136 or a similar course.
- Organizational skills and familiarity with computers sufficient to install new software and create a file system for the course.
- Knowledge of the Python language adequate to make alterations to an existing Python program.
- The ability to attend class.

Course Objectives

The course objective is to integrate key notions from algorithms, computer architecture, operating systems, compilers, and software engineering in one unified framework. This will be done constructively, by building a general-purpose computer system from the ground up. In the process, we will explore many ideas and techniques used in the design of modern hardware and software systems, and discuss major trade-offs and future trends. Throughout this journey, you will gain many cross-section views of the computing field, from the bare bone details of switching circuits to the high level abstraction of object-based software design.

Student Outcomes

Upon successful completion of this course, student will be better able to:

- apply knowledge of computing and mathematics appropriate to the programs student outcomes and to the discipline.
- analyze a problem, and identify and define the computing requirements appropriate to its solution.
- design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.
- function effectively on teams to accomplish a common goal.
- communicate effectively with a range of audiences.

Textbook

This semester I'll be using the following text. You need to purchase a copy.

The Elements of Computing Systems
Nisan and Schocken
MIT Press
2005

Online Resources

Please bookmark the following online resources immediately:

- with the exception of the textbook, all course material will be made available online, through the University of Montana's Moodle system,
- the textbook has a web site, and
- there is a Coursera Course for this text, and videos of lectures are available if you register (helpful if you have to miss a class).

Software

This course uses simulators to test the design of your hardware. They are written in Java and run on Windows, OSX, or Linux. Of course, you'll need the Java runtime environment installed on your computer. The software should be downloaded and configured according to the instructions here.

Course Format

This is mostly a hands-on course, structured around building a series of twelve hardware and software projects. Each project is accompanied by a design document, an API, an executable solution, a test script (illustrating what the module is supposed to do), and a detailed implementation plan (proposing how to build it). The projects are spread out evenly, so there will be no special pressure towards the semesters end.

Our time together in the classroom will be spent as follows.

Board work We will begin each lecture by randomly selecting students to go to the board and demonstrate their work. This will take about 20 minutes. Based on the student's performance I will award the same grade to both the student and the group. This is done so that the group is responsible for each member's understanding. Rubric for assessment is on the course Moodle. While presenting, students may ask their group two questions that have once sentence answers.

Lecture After board work, I will lecture for about 40 minutes. Beyond the background theory, I will focus on working examples that are similar to the assigned work. I will also complete problems that students were not able to.

Group problem solving Finally, groups will form and have the remainder of the time, about 20 minutes, to work on their assignments. Groups will be of four students, randomly selected. Groups will be created three times during the semester, at more or less equal time increments.

Meeting Times/Place

Times: Tuesday, Thursday 15:30–16:50
Place: Jeanette Rankin Hall 204

Final Exam Time and Place

Time: 1:10-3:10, Thursday, May 10
Place: Jeanette Rankin Hall 204

Grading Policy

Grading scale

A	94-100
A-	90-93
B+	87-89
B	83-86
B-	80-82
C+	77-79
C	73-76
C-	70-82
D+	67-69
D	63-76
D-	60-62
F	0-59

Students achieving the numerical scores above are guaranteed the associated letter grade. However, if average performance is low, I may decide to assign a higher letter grade for a lower score; e.g. a B+ for a numerical score of 84.

Students taking the course pass/no pass are required to earn a grade of D or better in order to pass.

Assessments and weights

The following assessments will be used and weighted according to the values in the table to determine final grades.

Component	**Description**	**Weight**
In-class problems	Problems worked on the board, by individual students. A rubric of assessment appears on the course web site.	20%
Group work	Assessment of individual student performance at the board will be given to each member of the group the student is in.	20%
Midterm I	Test of your knowledge of material presented in class and projects. Inclusive of material presented since first day of class.	15%
Midterm II	Test of your knowledge of material presented in class and projects. Inclusive of material presented after midterm I.	15%
Final Exam	Test of your knowledge of all material presented in class and projects. Inclusive of all material.	30%

Tentative schedule:

TUESDAY	THURSDAY
Jan 23rd **1** Course introduction and demonstration of tools, Introduction to Hardware Description Language (HDL), logic gates	25th **2** Combinational logic and the ALU (Arithmetic-Logic Unit)
30th **3** Combinational logic and the ALU (Arithmetic-Logic Unit)	Feb 1st **4** Sequential logic: memory hierarchy
6th **5** Sequential logic: flip-flop gates, registers, and RAM	8th **6** Machine language: instruction set, assembly and binary versions
13th **7** Machine language: assembly language programs	15th **8** Computer architecture I
20th **9** Computer architecture II	22nd **10** Assembler: language translation - parsing and symbol table
27th **11** Assembler: language translation - macro-assembly and construction of assembler	Mar 1st **12** Virtual machine I: modern virtual machines, stack based arithmetic, logical and memory access operations
6th **13** Virtual machine I: implementation of a VM from assembler language previously developed	8th **14** **Midterm Exam I**
13th **15** Virtual machine II: stack-based flow-of-control and subroutine call-and-return techniques, complete VM implementation	15th **16** High level language: introduce *Jack*, a simple high level language with Java like syntax
20th **17** High level language: trade-offs in language design and a simple, interactive game in *Jack*	22nd **18** Compiler I: context-free grammars and recursive parsing algorithms, building a tokenizer and parser for *Jack*.
27th *Spring Break*	29th *Spring Break*
Apr 3rd **19** Compiler I: syntax analyzer and XML output	5th **20** Compiler II: code generations, low-level handling of arrays and objects

TUESDAY	THURSDAY
10th **21** Compiler II: a full-scale compiler, generating VM code from XML produced previous week	12th **22** Operating system: design of OS/hardware and OS/software with regard to time/space efficiency of design
17th **23** Operating system: classic algorithms in OS design	19th **24** Time to catch up
24th **25** Wrap up/Course evaluation	26th **26** **Midterm Exam II**
May 1st **27** Study	3rd **28** **Final Exam**
8th **29**	10th **30**

Attendance Policy

Attendance will not be taken. Students absent when called up to work problems on the board will be given a grade of 0%. Another team member will be selected to go to the board at random. Students informing the instructor of a valid reason for missing class *in advance*, via email, will not be called to the board. Valid reasons include family emergencies and illness. I may ask for documentation of absence (doctors note, death certificate, etc.).

Academic Integrity

All students must practice academic honesty. Academic misconduct is subject to an academic penalty by the course instructor and/or a disciplinary sanction by the University. All students need to be familiar with the Student Conduct Code. I will follow the guidelines given there. In cases of academic dishonesy, I will seek out the maximum allowable penalty. If you have questions about which behaviors are acceptable, especially regarding use of code found on the internet or shared by your peers, please ask me.

Disabilities

Students with disabilities may request reasonable modifications by contacting me. The University of Montana assures equal access to instruction through collaboration between students with disabilities, instructors, and Disability Services for Students. Reasonable means the University permits no fundamental alterations of academic standards or retroactive modifications.