

University of Montana

## ScholarWorks at University of Montana

---

University of Montana Course Syllabi

Open Educational Resources (OER)

---

Spring 2-1-2004

### CS 232.01: Computer Architecture and Assembly Language Programming II

Jesse V. Johnson  
*University of Montana*

Follow this and additional works at: <https://scholarworks.umt.edu/syllabi>

**Let us know how access to this document benefits you.**

---

#### Recommended Citation

Johnson, Jesse V., "CS 232.01: Computer Architecture and Assembly Language Programming II" (2004).  
*University of Montana Course Syllabi*. 9380.  
<https://scholarworks.umt.edu/syllabi/9380>

This Syllabus is brought to you for free and open access by the Open Educational Resources (OER) at ScholarWorks at University of Montana. It has been accepted for inclusion in University of Montana Course Syllabi by an authorized administrator of ScholarWorks at University of Montana. For more information, please contact [scholarworks@mso.umt.edu](mailto:scholarworks@mso.umt.edu).

Computer Science 232  
Computer Architecture and Assembly Language  
Programming II  
Spring 2004 Syllabus

January 23, 2004

**Professor**

**Name:** Jesse Johnson  
**Office:** 417 Social Science  
**Telephone:** (406) 243-2356  
**Fax:** (406) 243-5139  
**Email:** johnson@cs.umt.edu  
**Web:** <http://cs.umt.edu/u/johnson>  
**Office Hours:** WF 15:00–16:00  
MTTh 11:00-12:00

**Online Assistance**

This semester I am using the University's Blackboard system to facilitate online learning. This is a system that I am new to. Some of the advantages appear to be:

- I will post all relevant course material there. This will include homeworks as well as solution keys.
- You will always be able to see your grade in the course.
- There is a mailing list facility that you are strongly encouraged to sign up for. It is here that I will post changes in assignments, exam dates, etc. I will also monitor the list and try to answer student questions about things that are not clear.

**Course web site:**

The text authors have also established some excellent online resources that can be explored at:

<http://books.elsevier.com/us/mk>

You'll have to look around, the specific link is too long.

## **Textbook**

*Computer Organization & Design*  
The Hardware Software Interface  
Second Edition  
David A. Patterson and John L. Hennessy  
1998, Morgan Kaufmann  
ISBN 1-55860-428-6

## **Prerequisite**

- Computer Science 231, or consent of instructor.

## **Course Objectives**

Course objective are as follows.

- Understand the role of performance in machine evaluation.
- Give insight into the design of processors.
- Teach the principles underlying design and development of computer systems for a variety of purposes.
- Trace the influences of important computing developments on the architecture of computer systems.
- Outline the architectural features of modern computers.
- Learn about, understand, and tell others about important aspects of modern computer architectures.
- Work with others to achieve learning objectives.

## **Meeting Times/Place**

**Times:** Monday, Wednesday, Friday 8:10–9:00 am

**Place:** Social Science 352

## **Final Exam Time and Place**

10:10 May 10th

The final exam time and place will be used to hand in project writeups, which are due at that time.

## Grading Policy

Grades of A-F will be assigned based on the following scale.

A	90-100
B	80-89
C	70-79
D	60-69
F	0-59

Grades will be based upon the following forms of evaluation.

Exercise	Number	Percentage of final grade
Homework	6	40%
Midterm Exam	2	30 %
Final Project	1	30%

I reserve the right to make changes to the grading policy that will be favorable to students grades.

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

## Attendance Policy

Attending lectures is not required, but highly recommended. I will be devoting a significant amount of time to discussing the assignments, as well as breaking down the class into groups to discuss assignments. The remainder of the time will be spent discussing the material that is found in the textbook.

## Late Assignments

Other than in exceptional circumstances, such as family emergencies *late homework will not be accepted.*

## Academic Integrity

As a student of the University of Montana, you are responsible for upholding all rules in the student conduct code. There are aspects of that code that are of particular importance in Computer Science courses. The electronic nature of the many assignments facilitates their dissemination. To be clear, from the student conduct code:

1. Plagiarism: Representing another person's words, ideas, data, or materials as one's own.
6. Submitting work previously presented in another course: Knowingly making such submission in violation of stated course requirements.

Of course, all other aspects of the student conduct code will be enforced as well. These are just the two that are commonly violated.

I will interpret these guidelines to the letter. Students found in violation will be penalized with the maximum punishment permitted in the student conduct code. That is to say, the matter will be handed over to the academic Dean and academic misconduct proceedings will take place.

In order to reconcile encouraged interaction between students and the academic misconduct policies, you must credit other students in your work. If, for example, you worked with others to develop some algorithm, or solve some homework problem, specifically mention those that you have worked with in the assignment that is handed in. Similarly, you must properly document and credit any online resources that you use.

Collaboration is encouraged. However, if you collaborate with others, the instructor reserves the right to question you about the material turned in. If it is evident that your understanding of what you turn in is weak, your grade will be lowered.

Students are to uphold a level of conduct becoming of adults. The use of profanity and abusive speech is not permitted under the student conduct code, and will not be tolerated in this course.

## **Schedule**

The following is a tentative schedule for the course. I reserve the right to modify the schedule to reflect the way the course is going with respect to completion of assignments, performance on exams, etc.

Week	Dates	Topics	Chapter(s)	Notes
1	Jan 26–30	Performance, Metrics	2	
2	Feb. 2–6	Performance, Benchmarking	2	
3	Feb. 9–13	The Processor, Finite State Machines	5	Feb 13, Last day to add/drop w/ refund
4	Feb. 16–20	Datapath and Control Multicycle CPUs	5	
5	Feb. 23–27	Pipelines, stalls	6	
6	Mar. 1–5	Branch prediction Superscalar pipelines	6	Midterm I
7	Mar. 8–12	Memory hierarchy, cache	7	Mar. 9 Drop Add transcript clear.
8	Mar. 15–19	Multi-level cache	7	I'm gone 3/15 and 3/19
9	Mar. 22–26	I/O Disks, Networks	8	I'm gone 3/22
10	Mar. 29–Apr. 2	Sync/Async, Polling	8	
11	Apr. 5–9	Multiprocessor architectures	9	I'm gone 4/5
12	Apr. 12–16	Shared memory vs. message passing	9	Midterm II
13	Apr. 19–23		Presentations	
14	Apr. 26–30		Presentations	Apr 30 Last day to withdraw.
15	May 3–7		Presentations	
16	May 10–14		Finals week	

### Disabilities

Students with disabilities are encouraged to meet with me to discuss any accommodations they require.

### Other Issues

- Turn off your cellphone, or set it to vibrate in class. Take the call outside the classroom.
- Do not talk in the classroom during lecture.