

CIS 253 Computer Org & Assembly Language

<http://jbwyatt.com>

wyatt@clarion.edu

wyattwyatt@gmail.com

PREREQS: CIS 202 INSTRUCTOR: Joseph Wyatt PLACE: 122 Becker Hall TIME: mwf 10:00 TERM: Fall, 2011

OBJECTIVE:

- This course has two goals:
1. Familiarize the student with basic computer organization.
 2. Provide an introduction to computer programming at the assembly language level.

The two goals are closely related in that to learn about the low-level details of the computer, it is necessary to understand assembly language. Likewise, to learn to program at an assembly level requires an understanding of basic computer organization. I will present material, set goals and evaluate achievement. I will recognize and attempt to match extra effort, but will not shoulder the responsibility for lack of effort. **You** are responsible for your performance.

OUTCOMES :

The successful student will:

- understand the difference between data and information;
- understand how unsigned & signed integers are stored internally within a computer;
- understand how floating point numbers are stored internally within a computer;
- understand how characters are stored internally within a computer;
- understand how instructions are stored internally within a computer;
- understand in detail how programs are executed;
- understand the basic architecture of a computer;
- understand the syntax of an assembly language such as Intel x86 ;
- create assembly language programs;
- demonstrate the ability to input data into an assembly language program;
- demonstrate the ability to output information from an assembly language program;
- demonstrate the ability to repeat statements in a loop within an assembly language program;
- demonstrate the ability to make decisions within an assembly language program;
- demonstrate the ability to create separate functions or methods within an assembly language program;
- understand and demonstrate the integration of assembly language with a HOL such as C++;
- understand and demonstrate how to find and correct errors within a program using a debugger.

TEXT and MATERIALS (<http://jbwyatt.com/253/>) :

"8086 Microprocessor Emulator - <http://www.emu8086.com/>"

"Code: The Hidden language of Computer Hardware and Software" by Charles Petzold, Microsoft Press, 2000.

Development will use the emu8086 emulator and the Visual C++ compiler. Both can be downloaded for free. A flash drive can also be very useful. Finally, *you will need much persistence and careful attention to detail - this is not an easy course...*

ELECTRONIC ACCESS:

My E-mail address is: wyatt@clarion.edu or wyattwyatt@gmail.com

My Web URL is <http://jbwyatt.com/>

OFFICE INFORMATION: (<http://jbwyatt.com/advisee.html>)

My office is in 141 Becker Hall. My office telephone is (814) 393-2643 - feel free to leave a message. Come see me! Office hours are as posted, but other hours can be arranged.

Please defer personal conversations and smalltalk until after class as it annoys other students and bugs the heck out of me.

TOPICS / SCHEDULE (42 classes) (<http://jbwyatt.com/253/>)

First third (14 classes):

1. Course Intro
 2. HW Intro
 3. Emu 8086
 4. Emu 8086 References
 5. Representation: Base 10, 2, 16, 8
 6. Representation: Unsigned Binary Numbers
 7. Representation: Signed Binary Numbers
 8. Representation: IEEE 754 Floating Point
 9. Registers, RAM & Data
 - A. Assembly Language Intro
 - B. MOV, ADD and SUB
 - C. Summary & Review
- » Test 1: web notes, lectures, reading

Copying code is cheating. Allowing others to copy your code is cheating.

You must protect your intellectual property as you protect your personal property - with all reasonable measures.

Second third (14 classes):

1. EMU Library
 2. Logical Operators, XCHG
 3. Labels & Loops
 4. More Labels & Loops
 5. Flags and Math Instructions
 6. Flags, CMP, TEST, Conditional Jumps
 7. Memory Addresses, Pointers
 8. Procedures
 9. Procedures in Emu
 - A. Devices & Ports
 - B. I/O Ports & Interrupts
 - C. Summary & Review
- » Test 2: web notes, lectures, reading

Last third (14 classes+1):

1. C++ and Assembly
 2. Stacks: Push and Pop
 3. Creating and Navigating a Maze
 4. Debugging Code
 5. More Math: DIV, SHR, MUL, SHL
 6. Architecture 1 & 2: Basics & Digital Logic
 7. Architecture 2: Digital Logic
 8. Breadboard SW: Logic Gates
 9. Architecture 3: Building a CPU
 - A. Performance
 - B. Speed & Timing
 - C. Summary & Review
- » Test 3: web notes, lectures, reading (*cumulative final*)

GRADES: (<http://jbwyatt.com/grades.htm>)

Grades are determined by your % score: 90+ = A ; 80 – 89 = B ; 70 – 79 = C ; 60 – 69 = D ; 59 & below = E.

Grades are determined as follows:

50%: TQ (tests and quizzes)

Three 100 point tests (make-up **only** with prior notice and excuse). Various 5 point quizzes: no make-up .

40%: AS (assignments)

6 - 8 programs - programs are worth between 5 and 20 points

10%: APE (attendance, participation, effort) come to class, bring book, notebook, pen, take notes

25 points starting score; -1 point for each unexcused absence / unprepared attendance: **MAX ABSENCES: 5.**

While in class you are expected to be attentive and to participate. Participation means constructive and informed (by way of doing the assignments and reading) discussion about the subject material.

SPECIAL NEEDS and CONSIDERATIONS:

Any special circumstances that may affect your performance in the class should be brought to my attention. Any student requiring accommodations for taking notes or tests should make arrangements to discuss their needs with me after the first class. Anything that's bothering you may affect your performance in class - please approach me and I'll try to help.