**COURSE:** CMPS 250 - Machine Organization and Assembly Language Programming
Department of Computing Sciences, University of Scranton

(Class meets in LSC 091 on M-W-F from 12:00 PM - 12:50 PM)

**DATE:** Spring 2017 (January 30, 2016 - May 19, 2017)

**INSTRUCTOR:** P. M. Jackowitz
**OFFICE:** LSC 192
**TELEPHONE:** (570) 941-6107
**EMAIL:** paul.jackowitz@scranton.edu
**WWW:** http://www.cs.scranton.edu/~jackowit/
**OFFICE HOURS:** As posted, and by appointment.

**Course Description:**
(Prerequisite: CMPS 144) An introductory study of the organization and architecture of computers undertaken through an exploration of various virtual machines. Programming at the assembly language level and interfacing with software components (primarily written in Java and C). Topics include representation of data and instructions, instruction sets, addressing modes, computer arithmetic, memory hierarchies, digital logic, microprogramming, pipelining, and parallel processing. (Essentially from Undergraduate Catalog 2016-2017)

**STUDENT LEARNING OUTCOMES:** At the completion of this course the successful student will have the ability to:

1. Explain the functional components and operation of a computer system in detail, with focus on the processor, memory and the system interconnect.
2. Explain the generic design of the processor's Control Unit and the Instruction Execution Cycle in detail.
3. Explain the processor's Arithmetic Logic Unit and in particular the integer operations of addition, subtraction, multiplication and division.
4. Explain and use common representations of data and instructions.
5. Read, understand and modify existing program components written in Intel Assembly Language, and develop additional ones.
6. Understand the syntax and semantics of any assembly language; in particular, have the ability, with appropriate reference materials, to fully understand and develop program components in any assembly language.
7. Read, understand and modify existing program components written in the C Programming Language, and develop additional ones.
8. Explain dynamic memory management in general and be experienced in the use of pointers in C, and memory addresses in assembly language.
9. Identify, explain and know the significance and use of common circuits used in computer systems: decoders, multiplexers, adders, flip-flops, etc.

(This page serves as our primary electronic communication tool for this course. You will use it to access required and optional course material, submit assignments for grading, and receive feedback. You must register to gain access to the functionality of this site.)

GRADING:

Tests:  (approximate date)
Week of March 6th  25%
Final Exam:  (comprehensive, schedule yet to be determined)  25%
Quizzes:  (as announced)  10%
Assignments:
  Programming, homework, etc.  40%

Attendance and Class Participation considered.
(Your attendance at all classes is expected. The accumulation of more than four absences may result in a diminished final course grade.)

PROCEDURES:

Lectures:
• please sit in the same seat for every class
• feel free to ask and answer questions, and to contribute to discussions

Tests and Quizzes:
• always announced in advance
• no make-ups will be given
• notice must be given if a test must be missed due to serious illness or emergency

Assignments:
• each student is required to do his/her own work
• discussions and cooperation among students is encouraged, but must not be to the point of representing someone else’s effort as your own
• academic dishonesty will be dealt with severely
• each assignment will have a specified due date, and a deadline (typically about 3 calendar days hence)
• work submitted after the due date is considered “late”, will be accepted for grading but may be assessed a penalty (depending upon how late it is, and whether or not worthwhile preliminary work had been submitted by the due date).
• work may not be submitted after the deadline, is considered too late, may not be accepted for grading, and may receive a grade of zero (depending upon whether or not worthwhile preliminary work had been submitted by the deadline).
• incomplete work generally will receive a grade much higher than zero
• work not submitted will receive a grade of zero