CSE 2120 Computer Organization and Machine Programming (3 credits)

Primary instructor: William Allen **Supporting instructor:** Shengzhi Zhang

Textbooks and references:

Kip R. Irvine, Assembly Language for Intel-Based Computers, Prentice Hall, 7th edition, 2015. (R)

Course information:

2014–2015 Catalog description: CSE 2120 Computer Organization and Machine Programming (3 credits) Introduces digital logic, computer arithmetic, instruction sets and the basic components of computer architecture. Covers arithmetic/ logic, control, memory and input/output units. Explores the relationship between computer architecture and machine language programming. Requires students to write programs in Intel assembly language. Prerequisites: CSE 1001.

Prerequisites by topic: Fundamentals of computer programming

Place in program: Required. Prerequisite for: CSE 2050 (2nd programming language, currently C++)

Course outcomes & related student outcomes: The student will be able to

1. Explain the organization of a typical computer system including the following:
   (a) Storage of data and instructions
   (b) Access and exchange data in memory and registers
   (c) Interfaces with input-output devices
   (d) Binary and hexadecimal number systems and integer arithmetic

   (1: Fundamental knowledge)
2. Understand digital logic and sequential circuits. (1: Fundamental knowledge)
3. Minimize logic expressions. (2: Scientific, computing, and engineering problem solving)
4. Use program debugging techniques. (3: Skillful use of tools)
5. Understand control flow and memory access via pointers. (2: Scientific, computing, and engineering problem solving)
6. Program basic building blocks of a computer system in machine and assembly languages and demonstrate low-level concepts related to computer programming. (3: Skillful use of tools)

Topics covered:

1. Introduction, overview of topics for this course (1.5 hours)
2. Data representation, binary number system, binary arithmetic, floating point (4.5 hours)

3. Boolean and Digital Logic: Boolean algebra, K-Maps, simplification of logic, gates, and circuits (6.0 hours)
4. Overview of architecture, components, and interconnections (3.0 hours)
5. Memory organization, addressing, Endian-ness, memory alignment, and performance (3.0 hours)
6. Introduction to performance enhancements, pipelines, memory cache, and I/O devices (3.0 hours)
7. Introduction to instruction sets: addressing modes, instruction formats, and examples (3.0 hours)
8. Assembly language: data declarations, arithmetic, control flow, and condition codes (6.0 hours)
9. Assembly language: shift/rotate, multiplication/division, and logic instructions (3.0 hours)
10. Implementation of memory: variables, constants, arrays, pointers, and indirection (3.0 hours)
11. Program design, assemblers, debugging, compiler-generated code, and file I/O (3.0 hours)
12. Procedure calls and modular design (3.0 hours)
13. Examinations (One on organization and one on assembly language) (3.0 hours)

Approved by: William Allen, Associate Professor & Shengzhi Zhang, Assistant Professor

Signature: _____ **Date:** 2/2[illegible]/2015

Signature: _____ **Date:** 2/2/2015