I have read and understood the policy on academic dishonesty ("cheating") as outlined in the syllabus for the FALL 2015 CSCI 2450 course. In particular, I understand that copying or providing to other students, in whole or in part, solutions (text or source code) to CSCI 2450 assignments from any source (including work done by former and current CSCI students, other humans, animals, zombies, materials downloaded from the Internet, etc.) not directly sanctioned by Aleksandar Zoranic is not acceptable. I understand that all work must be exclusively my own, with the exception of any team projects, for which I am allowed to collaborate with my assigned partner.

THERE IS NO FLEXIBILITY IN THIS POLICY. IF YOU CHEAT, YOU FAIL, AND YOUR ACADEMIC CAREER IS LIKELY TO BE PREMATURELY TERMINATED. THE "REASON" YOU CHEATED IS IMMATERIAL. This policy applies equally to students “transmitting” or “receiving” answers.

Print your name: _________________________________________

Sign your name: __________________________________________

Date: __________________________________________________________

No grades will be assigned to your work until you sign and hand in this agreement.
CSCI 2450-001: Machine Structure & Assembly Language Programming
Fall 2015 Course Syllabus
Aleksandar Zoranic

Me: Office NSSAL Lab / Math 322
Office hours Tu/Thu 12:30PM – 2:30PM or By appointment.
Email azorani1@uno.edu

You: A student who has credit in CSCI 1060, 1201, 1203, 1205, or 1583, some basic programming skills.

Meeting: Math 118, 11:00am-12:15pm Tue/Thu

Textbooks: Assembly Language for Intel-Based Computers (7th Edition)
Kip Irvine Prentice Hall; 7th edition
ISBN: 978-0133769401
The link to author’s website which includes certain chapters of this book, supplemental files, code examples etc. is @
http://www.kipirvine.com/asm/

Grading: Midterm Examination (October 5-9, 2015) 25%
Comprehensive Final (December 7-11, 2015) 25%
Two Major Projects 30%
Lab/Programming Assignments 20%
DETAILS

ATTENDANCE: You must make an effort to attend every lecture. You are responsible for everything covered in lecture, regardless of whether or not the material is addressed in the textbooks or handouts.

My job is to try to make the material interesting and to teach you more than you would have learned from simply reading a few books. Your job is to make attendance a priority, to be prepared for class, and to actively participate.

ASSIGNMENTS: There will be significant laboratory/programming assignments in this course. You should consider the due date for each assignment to be a hard deadline. When the due date arrives, turn in what you have. I do give partial credit, but late submissions are not accepted. Submission procedures will be discussed in class when assignments are issued.

In addition to programming assignments, you may be issued homework assignments consisting of number of questions/problems.

SUBMISSION RULES:
All files should be compressed into a single zip file and named using the following naming scheme:
1. LastName_FirstName_CSCI2450_AssignmentName.zip.
2. An example of a project #2 that I would submit would be Zoranic_Aleks_CSCI2450_Project2.zip.
3. Submit your assignment to Moodle with the following subject line:LastName_FirstName_CSCI2450_AssignmentName(ex:Walker_Johnnie_CSCI2450_Homework3).
**TESTS:** There will be one midterm and one final. The final examination is comprehensive with an emphasis on material after the midterm. Any missed test will receive a grade of zero unless arrangements are made with me prior to missing the test.

**ONLINE CLASS MATERIALS:** My lecture slides are available via Moodle. Please try to view the slides online as much as possible and avoid printing them! The trees will love you. Note that the slides do not cover the entire content of the course. Additional reading materials and copies of assignments will also be made available on Moodle. Be sure to check the Moodle site frequently.

**CHEATING:** Don't cheat. If you do, I will catch you, and I will pursue the harshest possible penalties. All submitted work must be exclusively your own. Cheating is:

1) Copying, in whole or in part, the solutions of former students, current students, or any other organism, alive or dead. “Copying” includes transmission through email, the Web, smoke signals, Morse code, or any other means.

2) Obtaining solutions from the Internet or other archival sources.

3) *Looking* at a solution for more than three seconds is cheating. If you see something that looks like a solution to a CSCI 2450 assignment or examination, avert your eyes and run away as fast as you can.

If I suspect that you have submitted an assignment that was derived from another source, or that you gave away your solution, all offending parties will be asked to justify the similarities. If exchange of solutions, in whole or in part, occurred, you will receive an “F” in the course and
face potential expulsion from the university. I guarantee you. Cheating is not worth the consequences.

Discussing assignments at a high level for clarification, discussing problems concerning the computing equipment, and studying in groups for examinations is not cheating, but every word you type for programming and written assignments has better be your own!

**GRADING SCALE:** The following grading scale is used. I never curve. Grading in college courses is objective and based directly on your performance. Please don’t ask me to change your grade on an assignment unless you clearly deserve it and can demonstrate that this is the case.

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

**BONUS ASSIGNMENTS:** There may be an opportunity for each student to make up for the grades s/he made on either of the two exams by submitting an extra programming assignment. I will give out these assignments after the tests have been graded and returned to you. Expect bonus assignments to be harder than the programming assignments – after all they will contribute to your exam grade.

**STUDENT LEARNING OUTCOMES:** Each student is expected to master the proficiency in Intel Assembly Language, as well as low level computer architecture with a modest understanding of current reverse engineering techniques and technologies. Upon completion of this course, students will be expected to analyze, create and debug low level ASM programs, data models and structures.
CSCI 2450 Curriculum

Chapter 1 – Basic Concepts

• Welcome to Assembly Language
• Virtual Machine Concept
• Data Representation
• Boolean Operations

Chapter 2 - x86 Processor Architecture

• General Computer Concepts
• IA-32 Processor Architecture
• IA-32 Memory Management
• 64-bit Processors
• Components of an IA-32 Microcomputer
• Input-Output System

Chapter 3 - Assembly Language Fundamentals

• Basic Elements of Assembly Language
• Example: Adding and Subtracting Integers
• Assembling, Linking, and Running Programs
• Defining Data
• Symbolic Constants
• 64-Bit Programming
Chapter 4 - Data Transfers, Addressing, and Arithmetic

- Data Transfer Instructions
- Addition and Subtraction
- Data-Related Operators and Directives
- Indirect Addressing
- JMP and LOOP Instructions
- 64-Bit Programming

Chapter 5 – Procedures

- Stack Operations
- Defining and Using Procedures
- Linking to an External Library
- The Irvine32 Library
- 64-Bit Assembly Programming

Chapter 6 - Conditional Processing

- Boolean and Comparison Instructions
- Conditional Jumps
- Conditional Loop Instructions
- Conditional Structures
- Application: Finite-State Machines
- Conditional Control Flow Directives
Chapter 7 - Integer Arithmetic

- Shift and Rotate Instructions
- Shift and Rotate Applications
- Multiplication and Division Instructions
- Extended Addition and Subtraction
- ASCII and Unpacked Decimal Arithmetic
- Packed Decimal Arithmetic

Chapter 8 - Advanced Procedures

- Stack Frames
- Recursion
- INVOKE, ADDR, PROC, and PROTO
- Creating Multimodule Programs
- Java Bytecodes (optional)

Chapter 9 - Strings and Arrays

- String Primitive Instructions
- Selected String Procedures
- Two-Dimensional Arrays
- Searching and Sorting Integer Arrays
- Java Bytecodes: String Processing (optional topic)
Chapter 10 - Structures and Macros

• Structures
• Macros
• Conditional-Assembly Directives
• Defining Repeat Blocks

Chapter 11 - MS-Windows Programming

• Win32 Console Programming
• Writing a Graphical Windows Application
• Dynamic Memory Allocation
• x86 Memory Management

Chapter 12 - Floating-Point Processing and Instruction Encoding

• Floating-Point Binary Representation
• Floating-Point Unit
• x86 Instruction Encoding

Chapter 13 - High-Level Language Interface

• Introduction
• Inline Assembly Code
• Linking 32-Bit Assembly Language Code to C/C++