CSCE 2610 Assembly Language and Computer Organization
Syllabus – Fall 2018

Class Meetings: M/W 4 – 5:20 pm in Room B155 Discovery Park
Instructor: Hui Zhao
Office: F278
Office Hours: M/W 3-4 pm
Email: Hui.Zhao@unt.edu
TA: Xianwei Cheng (xianweicheng@my.unt.edu)

Course Objectives
Today’s professionals of every computing specialty need to understand both hardware and software in order to program efficiently. The focus of this course is to improve your understanding of hardware/software interface of parallel computers. This course will introduce the relationship between hardware and software and focus on the concepts that are the basis for current parallel computers. This course will use ARM and assembly language as a framework for understanding the fundamentals of computing and interactions between hardware and software.

Text Book

Grading Policy
Homework: 20%
Programming Assignment: 15%
Quiz: 10%
Midterm exam: 20%
Final exam: 30%
Class participation: 5%
Late Policy
Students are strongly encouraged to turn in any assignments on-time. Unless otherwise noted for a particular assignment, the following late policy holds. Late assignments will be penalized by subtracting 20% of the total achievable points of that deliverable, if turned in within the first 24 hours after the due date. Between 24 to 48 hours late turn in will result in a reduction of 50% of the total achievable points.

Certain deliverables may not have ANY LATE day, as announced. Late point reductions cannot be made up by later improvements.

Academic Integrity
Unless explicitly noted, all work is to be done on an individual basis. Any violation of the university's guidelines for academic integrity will result in no credit for the course and further disciplinary action.

Student Outcomes
The following are associated ABET Student Outcomes for this course:

CE Students
1. an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics.
6. an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions
7. an ability to acquire and apply new knowledge as needed, using appropriate learning strategies

CS
1. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program’s discipline.
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.

IT
1. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program’s discipline.