CS 254: Computer Organization and Assembly Language

1. Course number and name: CS 254: Computer Organization and Assembly Language

2. Credits and contact hours: 3.000 Credit Hours, 3.000 Lecture hours

3. Instructor or course coordinator: Bradley Kjell

4. Text book, title, author, and year
   
   No printed text book required.

   a. other supplemental materials
      
      o Course notes are available on line: [http://chortle.ccsu.edu/AssemblyTutorial/index.html](http://chortle.ccsu.edu/AssemblyTutorial/index.html) (26 Chapters, approx. 300 pages)
      
      o The course uses Blackboard/Vista for programming exercises, quizzes, grade book, and sample programs.

5. Specific course information

   a. brief description of the content of the course (Catalog Description):
      
      Concepts of assembler language, machine language, macro-instructions, subroutines, program checkout, interrupt structure of assemblers, and use of operating system. No credit given to students with credit for MATH 472.

   b. prerequisites or co-requisites: CS 151 or MATH 471

   c. indicate whether a required, elective, or selected elective course in the program: Required

6. Specific goals for the course

   Course outcome 1: The student will learn how data and instructions are represented with binary patterns.
   
   Course outcome 2: The student will learn basic computer architecture.
   
   Course outcome 3: The student will become proficient in basic assembly language programming and will gain an understanding of how high level programming languages are implemented as basic assembly language.
   
   Course outcome 4: The student will learn how a computer system is built out of ascending layers of abstraction.

   The department’s outcomes (a) an ability to apply knowledge of computing and mathematics, (b) an ability to analyze a problem, (c) an ability to design, implement, and evaluate a computer-based system, (i) an ability to use current techniques, (j) an ability to
apply mathematical foundations and (k) an ability to apply design and development principles in the construction of software systems are supported in the course outcomes.

7. Brief list of topics covered

1) Data Representation
   a) integer representation, unsigned and two’s complement
   b) binary arithmetic and overflow
   c) floating point representation, precision, floating point arithmetic
2) Basic architecture
   a) basic machine cycle
   b) memory, addresses, registers, ALU
   c) machine instructions
3) Assembly Programming
   a) integer operations, comparison operations, branching and looping
   b) extended assembly language
   c) subroutines and linkage, run-time stack
4) Abstraction
   a) structured programming
   b) virtual machine, virtual memory

8. Expected Performance Criteria

Students demonstrate their mastery of data representation through programming and written assignments. Mastery of basic computer architecture is utilized in programming a basic processor in machine language. Students write weekly MIPS assembly language programs of moderate length. Understanding of levels of abstraction is demonstrated in written answers to questions on midterms and the final. The final is comprehensive and based on past finals for this class. Student performance on the final is expected to be consistent with student performance in past years.