Prepared by the Department of Mathematics
Date of Departmental Approval: February 15, 2017
Date Approved by Curriculum and Programs: March 8, 2017 Effective: Fall 2017

1. **Course Number: CSC250**
   Course Title: Computer Organization and Architecture

   Description: Students acquire an understanding and appreciation of a computer system's functional components, their characteristics, performance, and interactions. Students evaluate computer architecture to develop programs that can achieve high performance through a programmer's awareness of parallelism and latency. In selecting a system to use, students analyze the tradeoff among various components, such as CPU clock speed, cycles per instruction, memory size, and average memory access time. Topics include digital logic, assembly language machine organization, and hardware-level C and assembly language programming.

2. **Student Learning Outcomes** (instructional objectives, intellectual skills)**:**
   Upon successful completion of this course, students are able to do the following:
   - write simple C library functions in Intel or ARM assembly language
   - produce an optimized gate level hardware logic schematic from a truth table and/or a Boolean algebra expression with as many as 4 inputs including "don't care" combinations
   - utilize embedded system software development techniques and tools including cross compilers, cross assemblers, hardware debug monitors, and diagnostic software

3. **Credit(s): 3**

4. **Satisfies General Education Requirement:** No

5. **Prerequisite(s):** CSC 130 (Computer Programming II: Java)

6. **Semester(s) Offered:** Fall

7. **Suggested General Guidelines for Evaluation:** Quizzes, Logic and Software Projects, Midterm, Final

8. **General Topical Outline** (optional)

   I. Computing Technology History
   II. Binary numbers, addition, subtraction
   III. Multiplication, division, floating point
   IV. Transistors, gates, and combinational logic (Prototyping)
   V. Combinational logic for arithmetic
   VI. Sequential logic and memory (Adder construction)
   VII. Memory technology (Register bit circuit)
   VIII. ISA, datapath, fetch-execute cycle, addressing modes
   IX. Control units, RISC ISA datapath
   X. ARM 32-bit ISA and assembly language
   XI. ARM control flow and addressing
   XII. Subroutines, stack, examples
   XIII. Introduction to C syntax, types, expressions, control structures, functions
   XIV. Pointers and parameter passing, mixing C and assembly
   XV. Programmed I/O, DMA
   XVI. Interrupt-driven I/O
   XVII. Comparison of ARM with other architectures
   XVIII. Performance (Vote!)
   XIX. Pipelines and hazards, Branch prediction
   XX. Superscalar Break
   XXI. Caches Virtual memory
   XXII. I/O buses and devices
   XXIII. Secondary storage