September 27, 2012

METROPOLITAN STATE COLLEGE *of* DENVER
Office of Academic Affairs

REGULAR COURSE SYLLABUS

School of: Letters, Arts and Sciences

Department: Mathematical and Computer Sciences

Prefix & Course Number: CS 2400 **Crosslisted With*:** _____

Course Title: Computer Organization 2

Check All That Apply: Required for Major: X Required for Minor: _____ Specified Elective:

Required for Concentration: _____ Elective: X Service Course: _____

Credit Hours: 4 (4+0)

Total Contact Hours per semester (assuming 15-16 week semester):

Lecture 60 Lab 0 Internship 0 Practicum 0 Other (please specify type and hours): 0

Schedule Type(s): L **Grading Mode(s):** L

Restrictions (Variable Topics Course): _____

Prerequisite(s): CS 1400, CS 1050, and MTH 1110 (or equivalent), all with a grade of "C" or better, or permission of instructor

Corequisite(s): none

Prerequisite(s) or Corequisite(s): _____

Banner Enforced:

Prerequisite(s): (CS 1400 FOR LEVEL UG WITH MIN. GRADE OF C OR CS 190C FOR LEVEL UG WITH MIN. GRADE OF C) AND (CS 1050 FOR LEVEL UG WITH MIN. GRADE OF C OR CS 1050 FOR LEVEL UG WITH MIN. GRADE OF T) AND (MTH 1110 FOR LEVEL UG WITH MIN GRADE OF C OR MTH 1110 FOR LEVEL UG WITH GRADE OF T OR MTH 1400 FOR LEVEL UG WITH MIN. GRADE OF C OR MTH 1400 FOR LEVEL UG WITH MIN. GRADE OF T OR MTH 1120 FOR LEVEL UG WITH MIN. GRADE OF C OR MTH 1120 FOR LEVEL UG WITH MIN. GRADE OF T OR MTH 1410 FOR LEVEL UG WITH MIN. GRADE OF C OR MTH 1410 FOR LEVEL UG WITH MIN. GRADE OF T)

Corequisite(s): _____

Prerequisite(s) or Corequisite(s): _____

Catalog Course Description:

The course presents the functional organization of computers, multicore and multithreaded processors, high performance storage, multiprocessor and multicomputer parallel architectures, and error detecting/correcting codes. Students learn assembly language programming and create software using a contemporary development environment.

APPROVED:

Department Curriculum Committee	1-24-13 Date
Department Chair **OR** Program Director	1-24-13 Date
Dean **OR** Associate Dean	1/24/13 Date
Associate VP, Academic Affairs	2/28/13 Date

*If crosslisted, attach completed Course Crosslisting Agreement Form

Prefix and Course Number: CS 2400 September 27, 2012

Required Reading and Other Materials will be equivalent to:
Clements, Alan. (2006). *The Principles of Computer Hardware, 4th edition.* New York, NY. Oxford University Press
On-line books and open-source software.

Specific, *Measurable* Student Behavioral Learning Objectives:
Upon completion of this course the student should be able to

1. Describe how a processor's control unit interprets a machine-level instruction – either directly or as a microprogram.
2. Explain processor pipelining.
3. Compare performance of various processors and systems.
4. Describe how superscalar and vector architectures use hardware redundancy to execute more than one operation per processor cycle.
5. Explain the concept of multiprocessing.
6. Examine how special-purpose graphics processors can accelerate performance.
7. Describe cache memory organization.
8. Describe high performance hard disk and networked attached storage.
9. Analyze virtual memory organization.
10. Discuss the various classes of instructions: data movement, arithmetic, logical, and flow control instructions.
11. Design, implement, and test stand-alone assembly-language programs.
12. Explain basic error detecting/correcting codes.

Detailed Outline of Course Content (Major Topics and Subtopics) **or Outline of Field Experience/Internship (experience, responsibilities and supervision)**

I. Functional Organization
   A. Microarchitectures - hardwired and microprogrammed implementations of the control unit.
   B. Instruction-level parallelism and processor pipelining
   C. Overview of superscalar and vector architectures
   D. Branch Prediction
   E. Processor and system performance
II. Multiprocessing
   A. Short vector processing (multimedia operations)
   B. Multicore and multithreaded processors
   C. Flynn's taxonomy: multiprocessor structures and architectures
   D. Special-purpose parallel graphics processors
III. High Performance Storage
   A. Cache memories - block placement,identification,replacement and write policies, cache coherency
   B. Virtual Memory
   C. Increasing disk reliability and performance: RAID hard disks
   D. Storage area networks and network attached storage
IV. Assembly Programming Project
   A. Instruction sets and memory addressing
   B. Assembly language: label arithmetic, macrodirectives
   C. Assembly language programming techniques
V. Error Detecting/Correcting Codes
   A. Parity error detecting/correcting codes
   B. Polynomial codes

Prefix and Course Number: CS 2400 September 27, 2012

Evaluation of Student Performance

A combination of the following:

1. Homework and Programming Assignments
2. Quizzes and Examinations
3. Final Examination

Program Student Learning Outcomes Assessment

Students should achieve the Program Student Learning Outcomes (Program SLOs) by the time of graduation. Each individual Program SLO is assessed by selecting one or more course SLOs that contribute to the evaluation of that one Program SLO.

1. Program SLO c: An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs.
   - Course SLO #1: Describe how a CPU's control unit interprets a machine-level instruction – either directly or as a microprogram.
   - Course SLO #2: Explain processor pipelining.
   - Course SLO #3: Compare performance of various processors and systems.
   - Course SLO #6. Examine how special-purpose graphics processors can accelerate performance.
   - Course SLO #7: Describe cache memory organization.

2. Program SLO i: An ability to use current techniques, skills, and tools necessary for computing practices.
   - Course SLO #3: Compare performance of various processors and systems.
   - Course SLO #4: Describe how superscalar and vector architectures use hardware redundancy to execute more than one operation per processor cycle.
   - Course SLO #5: Explain the concept of multiprocessing.
   - Course SLO #9: Analyze virtual memory organization.

3. Program SLO j: An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.
   - Course SLO #5: Explain the concept of multiprocessing.
   - Course SLO #6: Examine how special-purpose graphics processors can accelerate performance.
   - Course SLO #8: Describe high performance hard disk and networked attached storage.
   - Course SLO #12: Explain basic error detecting/correcting codes.

4. Program SLO k: An ability to apply design and development principles in the construction of software systems of varying complexity.
   - Course SLO #10: Discuss the various classes of instructions: data movement, arithmetic, logical, and flow control instructions.
   - Course SLO #11: Design, implement, and test stand-alone assembly-language programs.