January 9, 2006

METROPOLITAN STATE COLLEGE *of* DENVER
Office of Academic Affairs

REGULAR COURSE SYLLABUS

School of: Letters, Arts, and Sciences

Department: Mathematical and Computer Sciences

CIP Code: 11.9999

Prefix & Course Number: CS 3280 **Crosslisted With*:** _____

Course Title: Object-Oriented Software Development

Check All That Apply: Required for Major: _____ 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): Lecture **Grading Mode(s)**: Letter

Variable Topics Courses (list restrictions, including the maximum number of hours that can be earned**):

_____

** **NOTE: This information must be included in the course description.**

Restrictions (Variable Topics Course): _____

Prerequisite(s): CS 2050 with a grade of "C" or better, or permission of instructor

Corequisite(s): none

Prerequisite(s) or Corequisite(s): _____

Banner Enforced:

Prerequisite(s): _____
Corequisite(s): _____
Prerequisite(s) or Corequisite(s): _____

Catalog Course Description:
This is an upper division software development class that focuses on the object-oriented programming paradigm. Object-oriented-analysis, -design, and –development will be explored in some depth with emphasis on object definition, abstraction, polymorphism, encapsulation, and inheritance. Abstract class definitions are developed for a number of common objects and data structures, and derivative classes and subclasses are developed from these definitions. Students will develop a thorough understanding of an object-oriented programming language such as C++ or Smalltalk.

APPROVED:

Ruth G. Yarar — 1-17-06
Department Curriculum Committee — Date

[signature] — 11/19/06
Department Chair **OR** Program Director — Date

Hal Tamay — 1/31/06
Dean **OR** Associate Dean — Date

Linda S. Curran — 2/2/06
Associate VP, Academic Affairs — Date

*If crosslisted, attach completed Course Crosslisting Agreement Form

Prefix and Course Number: CS 3280 January 9, 2006

Required Reading and Other Materials will be equivalent to:
Applying UML & Patterns, 2nd ed, Larman, Pearson, 2001

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

1. Define object-oriented programming, -design, and –analysis, and identify the principle characteristics that differentiate it from other programming paradigms.
2. Define criteria for identifying a candidate object.
3. Identify and define the state variables associated with an object.
4. Identify and define the behavior of an object in a specified environment.
5. Create abstract and parent object classes for use in defining derived and descendant classes.
6. Develop a set of virtual methods for a line of related and descendant object classes.
7. Develop the functionality of overloaded operators for a complex object class.
8. Design, develop, and implement object-oriented applications of significant complexity.
9. Develop an effective exception handling scheme for an application system.
10. Design, develop and implement a library of objects and object classes.

Detailed Outline of Course Content (Major Topics and Subtopics):

I. Evolution of Programming Languages from FORTRAN to Object-Oriented Languages
II. Elements of the Object Model
  A. Abstraction
  B. Encapsulation
  C. Modularity
  D. Inheritance
  E. Polymorphism
III. Classes and Objects
  A. What Constitutes and Object
  B. Relationships Between Objects
    Links, Aggregation
  C. What Constitutes a Class
  D. Relationships Between Classes
    Association, Aggregation, Inheritance, Metaclass, Delegation
  E. Relationships Between Classes and Objects
IV. The Realization of classes and Objects in a Programming Language
  A. Introduction to the Language
  B. Implementing Classes and Objects in the Language
  C. Difficulties of Implementing Object-oriented Theory in a Language
V. Object-Oriented Analysis and Design
  A. Classification
  B. Notation
  C. The Process
  D. Designing for Reuse
  E. Pragmatics
VI. Testing Object-Oriented Software

Prefix and Course Number: CS 3280 January 9, 2006

Evaluation of Student Performance

1. Homework and Programming Assignments
2. Quizzes and Examinations
3. Final Examination
4. Research Papers and/or Book Reports
5. Oral Presentations
6. Significant Programming Projects

As determined by the instructor. Written communication skills will be applied in this course.