September 13, 2010

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: 11.0701

Prefix & Course Number: CS 2050 **Crosslisted With*:** _____

Course Title: Computer Science 2

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

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 1050 and MTH 1110 (or equivalent) each 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 course, a continuation of CS 1050, further emphasizes the concepts of the software development cycle and introduces the concept of an abstract data type (ADT). The topics covered include linked-lists, trees, stacks, queues, classes, recursion, and a variety of data representation methods. Further topics in software engineering and programming style as well as algorithms for sorting and searching are included.

APPROVED:

Department Curriculum Committee	Date 10/14/10
Department Chair OR Program Director	Date 10/14/10
Dean OR Associate Dean	Date 11/12/10
Associate VP, Academic Affairs	Date 3/4/11

*If crosslisted, attach completed Course Crosslisting Agreement Form

Prefix and Course Number: CS 2050 September 13, 2010

Required Reading and Other Materials will be equivalent to:

Koffman & Wolfgang. (2005). *Objects, Abstraction, Data Structures and Design Using Java,* Version 5, Wiley & Sons, ISBN: 471753882

Specific, *Measurable* Student Behavioral Learning Objectives:

Upon completion of this course the student should be able to:

1. Describe the concepts associated with an Abstract Data Type (ADT) and create appropriate ADT's for specific applications.
2. Implement an ADT as a class
3. Implement a linked list in a variety of ways including singly linked, doubly linked, circular, dummy head node.
4. Describe and contrast the various ways to implement a linked list.
5. Describe the stack as an ADT.
6. Implement a stack using an array or a linked list.
7. Describe the queue as an ADT.
8. Implement a queue using an array or a linked list.
9. Describe the list as an ADT.
10. Implement a list using an array or a linked list.
11. Describe general trees, binary trees, and binary search trees as ADTs.
12. Implement trees.
13. Describe various common applications of trees.
14. Write recursive procedures to perform preorder, inorder, and postorder tree traversals.
15. Determine the order of a simple algorithm.
16. Describe and implement various sorting algorithms, and be able to identify the order of complexity of their algorithms.
17. Describe and implement various searching algorithms, and be able to identify the order of complexity of their algorithms.
18. Identify and describe the elements of the software development cycle.
19. Write program documentation that includes appropriate preconditions and postconditions.

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

I. Software Engineering
   - A. Software development cycle
   - B. The characteristics of a good program
   - C. Programming style
   - D. Preconditions and postconditions
   - E. Loop invariants
   - F. Debugging
   - G. Introduction to the analysis of algorithms including Big-O notation

II. Abstract Data Type and Classes
   - A. Definition and examples of ADTs
   - B. Implementing ADTs as classes
   - C. Inheritance

III. Linked Lists
   - A. Singly linked lists
   - B. Doubly linked lists
   - C. Circular lists
   - D. Dummy head nodes

IV. Stacks and Queues
- A. The ADT stack
- B. Implementations of stacks
- C. The ADT queue
- D. Implementations of queues

V. Lists
- A. The ADT list
- B. Implementations of lists

VI. Trees
- A. General, binary and binary search trees
- B. Traversals of binary trees
- C. Heaps

VII. Applications
- A. Prefix, postfix, infix notation for expressions
- B. Translating infix to postfix notation
- C. Evaluating postfix expressions
- D. How recursion is implemented

VIII. Sorting
- A. Selection sort, insertion sort, Shell sort, quicksort, merge sort, heap sort, and tree sort with an algorithmic analysis of each
- B. Comparison of the methods

IX. Searching
- A. Sequential search
- B. Binary search
- C. Introduction to hashing
- D. Binary search trees and an overview of balancing techniques
- E. Comparison of the methods

Evaluation of Student Performance:

A combination of the following:
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

Written communication skills will be applied in this course.