CS - 4050 - Algorithms and Algorithm Analysis

z08. UG Course Modification #2 (Substantive College) (18-19)

General Catalog Information

Tracking: LAS 1617-47

Department* Mathematical and Computer Sciences, Department of

Status:* ☑ Active-Visible

Prefix:* CS

Course Number:* 4050

Course Type:* Computer Science

Course Title:* Algorithms and Algorithm Analysis

Transcript Course Title:* Algorithms & Algorithm Analysi

Equivalent/ Crosslisted? ☐ Equivalent ☑ Crosslisted

List all equivalent courses:

List all crosslisted courses: HON 4050

Specify Course Modifications:* pre- or co-requisites

Justification for Proposal:* The new prerequisite course, CS 3250, is intended to provide background in software development tools helpful for programming projects in this course.

Paste results from impact Report:*

Impact Report for CS 4050

Source: 2016-2017 Undergraduate Catalog	
Cross Listed Course(s):	HON 4050 - Algorithms and Algorithm Analysis
Programs	Computer Science Major, B.S.

Check All That Apply:*

☑ Required for Major
□ Required for Minor
□ Required for Concentration
□ Specified Elective
□ Elective

Credit Hours:* 4

Schedule Type:* Lecture

Grade Mode:* Letter

Lecture: 60

Lab:

Internship:

Practicum:

Other:

Additional Student Work Hours per course: 120

Variable topics umbrella course:* ● No ○ Yes

If yes, number of credits/ repeats allowed

Specified repeatable course:* ● No ○ Yes

If yes, number of credits/ repeats allowed

Prerequisite(s): CS 3240, CS 3250, and 4 additional credits of upper division CS courses all with grades of "C" or better, or permission of instructor. MTH 3210 is recommended.

Corequisite(s):

Prerequisite(s) and/or Corequisite(s):

Banner Prerequisite(s):

Banner Corequisite(s):

Banner Prerequisite(s) and/or Corequisite(s):

Level

Class

Program/Major	
Student attribute	
Catalog Course Description:*	The emphasis of this course is on the design, analysis, and evaluation of efficient algorithms for a wide variety of computing problems.
Required Reading and Other Materials will be equivalent to:	Algorithms in C++, Sedgewick, Addison Wesley, 1992. ISBN: 0-201-51059-6
Specific, Measurable Student Behavioral Learning Objectives:	Upon completion of this course the student should be able to: 1. Analyze algorithms 2. Use a recurrence relation to analyze a recursive algorithm. 3. Remove recursion from an algorithm. 4. Discuss and implement random number generator. 5. Implement a linked list using arrays. 6. Describe the shortcoming of brute-force algorithms. 7. Implement a greedy algorithm to solve an appropriate problem. 8. Implement a divide-and-conquer algorithm to solve an appropriate problem. 9. Discuss alternatives to exhaustive search. 10. Use backtracking to solve an appropriate problem (e.g.: navigating a maze). 11. Describe various heuristic problem-solving methods. 12. Implement a string searching algorithm. 13. Use pattern matching to analyze substrings. 14. Use dynamic programming to solve an appropriate problem. 15. Discuss various methods of parsing. 16. Implement distributed algorithms.

Detailed Outline of Course Content (Major Topics and Subtopics) or Outline of Field Experience/ Internship

I. Algorithm Analysis
A. Asymptotic analysis of upper and average complexity bounds
B. Identifying differences among best, average, and worse case behaviors
C. Big O, little o omega, and theta notation
D. Standard complexity classes
E. Time and space tradeoffs in algorithms
F. Using recurrence relations to analyze recursive algorithms
II. Removal of Recursion
III. Random Number Generators
IV. Array Implementation of a Linked List
V. Algorithm Strategies
A. Brute-force
B. Greedy
C. Divide-and-Conquer
D. Exhaustive Search
E. Backtracking
F. Branch-and-bound
G. Heuristics
H. String Searching
1. Brute Force
2. Knuth-Morris-Pratt Algorithm
3. Boyer-Moore Algorithm
I. Pattern matching
1. Pattern Matching Machine
J. Dynamic programming
1. Knapsack
2. Matrix-chain multiplication
3. Optimal Binary Search Trees
VI. Parsing
A. Introductory Concepts
B. Top-Down and Bottom-Up Parsing
VII. Distributed Algorithms
A. Consensus and Election
B. Termination Detection
C. Fault Tolerance
D. Stabilization

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.

Distribution of Credit Hours

(4 + 0)