COURSE NUMBER: Ve281	**COURSE TITLE:** Data Structures and Algorithms
CREDIT: 4	**PREREQUISITES:** Ve203 and Ve280
TEXTBOOKS/REQUIRED MATERIAL: "Data Structures and Algorithm Analysis," C. Shaffer	**INSTRUCTOR:** Weikang Qian **DATE OF PREPARATION:** Oct. 9, 2012 **DATE OF UC APPROVAL:** Oct. 30, 2013
INSTRUCTOR(S): Weikang Qian	**SCIENCE/DESIGN:** n/a
CATALOG DESCRIPTION: Introduction to algorithm analysis and O-notation; Fundamental data structures including lists, stacks, queues, priority queues, hash tables, binary trees, search trees, balanced trees and graphs; searching and sorting algorithms; recursive algorithms; basic graph algorithms; introduction to greedy algorithms and divide and conquer strategy. Several programming assignments.	**COURSE TOPICS:** 1. Asymptotic analysis and big-Oh notation (1.5 lectures) 2. Pointers, arrays, and dynamic memory management (1.5 lectures) 3. Linked list and its optimization (2 lectures) 4. Stacks and queues (1 lecture) 5. Priority queues (1 lecture) 6. Hash tables (2 lecture) 7. Trees (5 lecture) 8. Graphs and basic graph algorithm (5 lectures) 9. Searching and sorting algorithms (2 lectures) 10. Greedy algorithm (1.5 lectures) 11. Divide and conquer strategy (1.5 lectures) 12. Dynamic programing (2 lectures)
COURSE STRUCTURE/SCHEDULE: Lecture: twice per week, 90 minutes each.	
COURSE OBJECTIVES [Course Outcomes in brackets]	1. To teach students common data structures and algorithms. [1, 2, 4, 7] 2. To teach students how to analyze the time/space complexity of data structures and algorithms. [3, 5] 3. To provide students with experience to solve real problems using the existing data structures and algorithms. [4, 5, 6, 7]
COURSE OUTCOMES [Program Outcomes in brackets]	After completing Ve281, students should be able to: 1. Get familiar with common data structures and algorithms. [a, c] 2. Know how to implement these data structures and algorithms. [a] 3. Know how to analyze the time/space complexity of these algorithms. [a] 4. Take a problem and consider various possible data structures and algorithms for solving it. [a, c, e] 5. Select the data structure and the algorithm that are most efficient in solving the problem. [b, e] 6. Write code to solve the problem using the selected data structures and algorithms. [b, e, k] 7. Be able to quickly design, implement, test and debug a large scale project independently (1000+ lines of code). [b, e, k]
ASSESSMENT TOOLS [Course Outcomes in brackets]	Written Assignments [1, 2, 3, 4, 5, 6] Programming Projects [1, 2, 3, 4, 5, 6, 7] Midterm and Final Exam [1, 2, 3, 4, 5, 6]