

## CSCE 350: Data Structures and Algorithms

1. Course number and name: CSCE 350: Data Structures and Algorithms
2. Credit: 3-hrs; Contact: 3 lectures of 50 minutes each or 2 lectures of 75 minutes each per week
3. Instructor: Fall 2010: Jason O'Kane  
Spring 2011: Song Wang
4. Text book: Anany V. Levitin, *Introduction to the Design and Analysis of Algorithms*, Second Edition, Addison Wesley, Boston, MA, 2007.
5. Specific course information
  - a. Catalog description: Techniques for representing and processing information, including the use of lists, trees, and graphs; algorithm design strategies; analysis of algorithms; sorting, searching, and hashing techniques.
  - b. Prerequisites: CSCE 146 and MATH 374
  - c. Required in all curricula
6. Specific goals for the course
  - a. Specific outcomes of instruction:
    1. Describe formal analysis measures.
    2. Describe the relevance of abstraction to problem solving.
    3. Analyze and use lists, trees, and graphs.
    4. Apply common algorithm design techniques such as brute force, divide-and-conquer, decrease-and-conquer, transform-and-conquer, dynamic programming, and the greedy technique.
    5. Analyze algorithms.
    6. Use appropriate data structures.
  - b. Relation of course outcomes to Student Outcomes: CE: see page 2; CS & CIS: see page 3
7. Topics covered and approximate weight (14 weeks, 3 hours/week, 42 hours total)
  1. Structured programming, stacks, queues, lists (3 hours)
  2. Determining the Running Time of Programs, Order of Magnitude Analysis (6 hours)
  3. Brute force (3 hours)
  4. Divide-and-Conquer (4 hours)
  5. Dynamic Programming (6 hours)
  6. Transform-and-Conquer (4 hours)
  7. The Greedy Technique (3 hours)

8. Decrease-and-Conquer (3 hours)
9. Graphs (3 hours)
10. Reviews and exams (4 hours)

c.

## Computer Engineering

### Relation of Course Outcomes to EAC Student Outcomes\*

Course Outcomes (CE)	Student Outcomes											
	(a) apply knowl edge of mathe matics , scienc e, and engine ering	(b) design and condu ct experi ments, ... interpr et data	(c) design a syste m, comp onent, or proces s to meet desire d needs ...	(d) functi on on multid iscipli nary teams	(e) identif y, formu late, and solve engine ering proble ms	(f) an unders tandin g of profes sional and ethical respon sibilit y	(g) comm unicat e effecti vely	(h) the broad educat ion to unders tand the impac t of engine ering soluti ons ...	(i) a recogn ition of the need for, and an ability to engag e in life- long learn ing	(j) a knowl edge of contem pora ry issues	(k) use the techni ques, skills, and moder n engine ering tools ....	(CE) demo nstrate knowl edge of discret e mathe matics [CE]
Criteria	a	b	c	d	e	f	g	h	i	j	k	CE
1. Describe formal analysis measures.	3							2				3
2. Describe the relevance of abstraction to problem solving.		2	2	1	3			2		2		
3. Analyze and use lists, trees, and graphs.	3	1										3
4. Apply common algorithm design techniques such as brute force, divide-and-conquer, decrease-and-conquer, transform-and-conquer, dynamic programming, and the greedy technique.	3		1		2						1	3
5. Analyze algorithms.	3											3

\* 3 = major contributor, 2 = moderate contributor, 1 = minor contributor; blank if not related

d.

## Computer Science & Computer Information Systems

Relation of Course Outcomes to CAC Student Outcomes\*

<b>Course Outcomes (CS &amp; CIS)</b>	<b>Student Outcomes</b>											
	<b>All</b>									<b>CS</b>		<b>CIS</b>
	(a) apply knowledge of computing and mathematics appropriate to the discipline	(b) analyze a problem, and identify and define the computing requirements ...	(c) design, implement, and evaluate a computer-based system, ...	(d) function effectively on teams to accomplish a common goal	(e) An understanding of professional, ethical, legal, ... responsibilities	(f) communicate effectively with a range of audiences	(g) analyze the local and global impact of computing on ... society	(h) Recognition of the need for ... continuing professional development	(i) current techniques, skills, and tools necessary for computing practice	(j) apply mathematical foundations, algorithmic principles, and CS theory ...	(k) apply design and development principles	(l) An understanding of those processes that support the information systems environment.
Criteria	a	b	c	d	e	f	g	h	i	j	k	l
1. Describe formal analysis measures.	2	3								3	3	
2. Describe the relevance of abstraction to problem solving.	3	3	2						3	3		2
3. Analyze and use lists, trees, and graphs.	2	1							2	3		
4. Apply common algorithm design techniques such as brute force, divide-and-conquer, decrease-and-conquer, transform-and-conquer, dynamic programming, and the greedy technique.	3	3							3	3	3	
5. Analyze algorithms.	3	3	2							3	3	

\* 3 = major contributor, 2 = moderate contributor, 1 = minor contributor; blank if not related