CSCE 145: Algorithmic Design I

1. Course number and name: CSCE 145: Algorithmic Design I

2. Credit: 4-hrs; Contact: 2 lectures of 50 minutes each; 2 labs of 1 hour and 55 minutes each per week

3. Instructor: Fall 2010: Michael N. Huhns, Jose M. Vidal
   Spring 2011: Michael N. Huhns

4. Text book: Walter Savitch and Frank Carrano, *Java: Introduction to Problem Solving and Programming (5th Edition)*, Prentice Hall, 2009, ISBN-10: 0136072259.

5. Specific course information
   a. Catalog description: Problem-solving, algorithmic design, and programming. Open to all majors.
   b. Prerequisites: Placement in MATH 141 or grade of C or better in MATH 115
   c. Required in all curricula

6. Specific goals for the course
   a. Specific outcomes of instruction are that students will be able to:
      1. Solve problems using a computer
      2. Read and design algorithms
      3. Design data structures
      4. Demonstrate the ability to use a software development environment to construct, execute, test, and debug software
      5. Demonstrate the ability to program a computer in a high-level language
   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, 4 hours/week, 56 hours total)
   1. Introduction to programming tools (1 hour)
   2. Primitive data types, including strings (4 hours)
   3. Flow of control (5 hours)
   4. Classes, methods, and encapsulation (7 hours)
   5. Method overloading and constructors (4 hours)
   6. Arrays (4 hours)
   7. Inheritance and polymorphism (6 hours)
   8. Exceptions and exception handling (4 hours)
   9. Input/output using streams and files (5 hours)

10. Graphical user interfaces (4 hours)
11. Applications such as robotics, digital signal processing, and website animation (7 hours)
12. Reviews and Examinations (5 hours)

c.

Computer Engineering

Relation of Course Outcomes to EAC Student Outcomes*

Course Outcomes (CE)	Student Outcomes											
	(a) apply knowledge of mathematics, science, and engineering	(b) design and conduct experiments, … interpret data	(c) design a system, component, or process to meet desired needs …	(d) function on multidisciplinary teams	(e) identify, formulate, and solve engineering problems	(f) an understanding of professional and ethical responsibility	(g) communicate effectively	(h) the broad education to understand the impact of engineering solutions …	(i) a recognition of the need for, and an ability to engage in life-long learning	(j) a knowledge of contemporary issues	(k) use the techniques, skills, and modern engineering tools ….	(CE) demonstrate knowledge of discrete mathematics [CE]
Criteria	a	b	c	d	e	f	g	h	i	j	k	CE
1. Solve problems using a computer	2	1	2		3	1		1	2	1	3	1
2. Read and design algorithms	1	2	3		1						2	1
3. Design data structures	1	1	3		1						2	1
4. Demonstrate the ability to use a software development environment to construct, execute, test, and debug software	1	1	2		2						3	
5. Demonstrate the ability to program a computer in a high-level language	1		2		1						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*

Course Outcomes (CS & CIS)	Student Outcomes											
	All									CS		CIS
	(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	(j) An understand processes that support the information systems environment.
Criteria	a	b	c	d	e	f	g	h	i	j	k	j
1. Solve problems using a computer	2	3	2				1	1	2	2	3	2
2. Read and design algorithms	1	3	2						2	2	3	
3. Design data structures	1	2	2						2	2	3	
4. Demonstrate the ability to use a software development environment to construct, execute, test, and debug software	1	2	3						3		2	1
5. Demonstrate the ability to program a computer in a high-level language	1	1	2						3	1	2	

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