CSCE 145: ALGORITHMIC DESIGN I

Catalog Course Description:
145—Algorithmic Design I. (4) (Prereq: Placement in MATH 141 or grade of C or better in MATH 115) Problem-solving, algorithmic design, and programming. Three lectures and two laboratory hours per week. Open to all majors.

Prerequisite(s) By Topic:
Precalculus mathematics

Textbook(s) and Other Required Material:
Walter Savitch, *Java: An Introduction to Problem Solving and Programming*, 4th edition, Pearson Prentice Hall, Upper Saddle River, NJ, 2005.

Computing Platform: Windows XP; JDK 1.5; Eclipse IDE

Course Objectives: {Assessment Methods Shown in Braces}
1. Understand how to solve problems using a computer. {tests}
2. Understand how to read and design algorithms. {tests}
3. Understand how to design data structures {tests}
4. Demonstrate the ability to use a software development environment to construct, execute, test, and debug software. {laboratory projects}
5. Demonstrate the ability to program a computer in a high-level language. {programming assignments, laboratory projects, tests}

Topics Covered:
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 (6 hours)
9. Input/output using streams and files (6 hours)
10. Vectors and linked lists (4 hours)
11. Graphical user interfaces (4 hours)
12. Reviews and Examinations (5 hours)

Syllabus Flexibility: Low. The Undergraduate Committee approves the choice of textbook and syllabus.

Laboratory Projects:
Twelve programming projects, covering the range of topics for the course, are developed during the weekly laboratory sessions. Ten software systems with more extensive capabilities are developed as part of homework assignments.
Other Course Work: At least two major in-class tests, two laboratory exams, and a final examination.

Assessment Methods:

1. Written examinations
2. Written programming projects as homework
3. Laboratory programming projects
4. On-line laboratory examinations

Relationship of Course to Program Outcomes:

The contribution of each course objective to meeting the program outcomes is indicated with the following scale: 3 = major contributor, 2 = moderate contributor, 1 = minor contributor. Blank if not related.

Course Objectives	Program Outcomes										
	1. Logic & Math	2. Computing Fundamentals	3. Apply Computing Principles	4. Work on Teams	5. Communicate Effectively	6. Liberal Arts & Soc. Sciences	7. Basic Science & Lab Methods	8. Learn New Tools & Processes	9. Employed on Graduation	10. Application Area	11. Electronics & Digital System Design
1. Solve problems using a computer		3	2						1		
2. Read and design algorithms	1	3	2						1		
3. Design data structures		3	2						1		
4. Use a software development environment to construct, execute, test, and debug software		1	1					3	2	1	
5. Program a computer in a high-level language		3	1		1			1	2		

Estimated Computing Category Content (Semester hours):

Area	Core	Advanced	Area	Core	Advanced
Algorithms	1		Data Structures	1	
Software Design	1		Programming Languages	1	
Computer Architecture					

Estimated Information Systems Category Content (Semester hours):

Area	Core	Advanced	Area	Core	Advanced
Hardware and Software	1		Networking and Telecommunications		
Modern Programming Language	2		Analysis and Design	1	
Data Management			Role of IS in an Organization		
Quantitative Analysis			Information Systems Environment		

Oral and Written Communication:
Development of readable and well documented programs.

Social and Ethical Issues: None

Theoretical Content: None

Analysis and Design:
Programming and problem solving by computer

Class/Laboratory Schedule:
Lecture: 3 periods of 50 minutes or 2 periods of 75 minutes per week
Laboratory: 1 period of 2 hours per week

Course Coordinator: Michael Huhns

Modification and Approval History
Initial description, April 1999
Revised, November 2000
Revised, February 2005
Revised June 2005 by Caroline Eastman to modify format and add Information Systems categories
Revised, June 2005 by Michael Huhns to incorporate several minor changes, including a new edition of the textbook