COURSE DESCRIPTION

Department and Course Number: CSCI 112

Course Title: Computer Science II

Current Catalog Description: Continuation of CSCI 111 with emphasis on computer programming as a systematic discipline. The topics include data structures, abstract data types, algorithm design and analysis, and programming methods and standards.

Total Credits: 3 hours

Coordinator: Cynthia Brown Zickos, Instructor of Computer and Information Science

Textbook: Lambert, Kenneth A. and Martin Osborne. *Java, A Framework for Program Design and Data Structures*. Brooks/Cole Thompson Publishing, Inc., 2000.

Other required materials: Two diskettes

References: http://www.olemiss.edu/~cbzickos/cs112/

Course Goals: To continue developing a disciplined approach to the design, coding, and testing of programs written in a block structured high-level language. To teach the use of data abstraction using as examples data structures other than those normally provided as basic types in current programming languages; for examples linked lists, stacks, queues, and trees. To introduce and analyze well-known searching and sorting algorithms. To teach the principles of recursion. To provided a foundation for the further studies in computer and information science.

Prerequisites by Topic: Topics of Computer Science I (CSCI 111)

Corequisites: MATH 261 or MATH 267

Major Topics Covered in the Course:

1. Object-oriented programming techniques. (3 hours)
2. Advanced java syntax. (2 hours)
3. Advanced array processing. (1 hour)
4. Linked lists processing. (3 hours)
5. Measuring efficiency. (2 hours)
6. Searching and sorting techniques. (2 hours)
7. Stack ADT. (1 hours)
8. Queue ADT. (1 hours)
9. List ADT. (1 hour)
10. Recursion. (3 hours)
11. Binary search tree ADT. (2 hours)
12. Heap ADT. (2 hours)
13. Priority queue ADT. (1 hour)
14. Tests and quizzes. (3 hours)

Laboratory projects : Students are required to turn in approximately 9 programming assignments. The first assignment requires 1 week. Other assignments require from 7 to 10 days.

1. Course introduction web exercise.
2. Program using advanced array processing and linked lists.
3. Program utilizing the student's knowledge of the Stack ADT.
4. Program utilizing the student's knowledge of the Queue ADT.

5. Program utilizing the student's knowledge of the List ADT. (sometimes included)
6. Program utilizing the student's knowledge of the Recursion. (may be incorporated into other labs)
7. Program utilizing the student's knowledge of the Binary Search Tree ADT.
8. Program utilizing the student's knowledge of the Heap ADT.
9. Program utilizing the student's knowledge of the Priority Queue ADT.

Estimate of ABET/CAC Category Content:

	CORE	ADVANCED		CORE	ADVANCED
Data Structures	1		Computer Organization and Architecture		
Algorithms	1		Concepts of Programming Languages	1	
Software Design					

Oral and Written Communications:

Not a significant focus of the course.

Social and Ethical Issues:

Not a significant focus of the course.

Theoretical Content (Foundations):

For every ADT and its associated algorithms, there is discussion of the measurement of efficiency with respect to time and space using Big O notation.

Problem Analysis and Solution Design:

There are several opportunities in our class to study the progressive development of an algorithm and/or data structure from an implementation that "works" to an implementation that is efficient. We are basically studying the classic ADTs and algorithms, so the student does not focus on analyzing new problems or synthesizing new solutions, except to a small degree on their programming assignments.