COURSE DESCRIPTION

Dept., Number	CSci 112	Course Title	Computer Science II
Semester hours	3	Course Coordinator	Cynthia B. Zickos, Instructor

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.

Textbook

Malik, D.S. *Java Programming: Program Design Including Data Structures,* Thompson Course Technology, Boston, Massachusetts, 2006. ISBN: 1–4188–3540–4.

References

The URL for the course website is at http://www.olemiss.edu/~cbzickos/cs112. This includes the syllabus and other related material.

Course Outcomes

Upon successful completion of this course, the students can:

1. Use proven software engineering techniques and methodologies to design, implement, test and document a computer program that solves a basic computing problem using:
   a. Two-dimensional arrays
   b. Singly linked lists
   c. Stacks, queues, lists, binary search trees
   d. Simple searching and sorting techniques
   e. Recursion
2. Choose the most efficient computing solution when using Big-O notation as a basis for comparison.

Relationship between Course Outcomes and Program Outcomes

1. Use proven software engineering techniques and methodologies to design, implement, test and document a computer program that solves a basic computing problem using: Two-dimensional arrays, Singly linked lists, Stacks, queues, lists, binary search trees, Simple searching and sorting techniques, and Recursion. Outcome i.
2. Choose the most efficient computing solution when using Big-O notation as a basis for comparison. Outcomes a, b, c, j.

Prerequisites by Topic

Fundamental programming concepts and skills (CSci 111)

Major Topics Covered in the Course

1. Two dimensional arrays.
2. Linear and binary search using arrays.
3. Introduction to sorting: selection and merge sorting.
4. Basic introduction to Big O notation.
5. Singly linked lists.
6. Array based and linked-list based implementations of the:
   a. List ADT
   b. Stack ADT
   c. Queue ADT
7. Introduction to Binary Search Trees
8. Applications of Lists, Stacks, Queues, and Binary Search Trees.
9. Recursion.
10. User defined classes.
11. Basic methods such as constructors, accessors, mutators, and toString.
12. Scope issues.
13. Non-static vs. static methods and data.
14. Class and instance methods and data.
15. Generics only as necessary.
16. Exception handling only as necessary.
17. Inheritance and polymorphism only as necessary.

-

Assessment Plan for the Course

A comprehensive, 30-question exam constructed by a faculty committee is administered to each offering of CSci 112. Student performance is analyzed question-by-question to identify needed adjustments in the textbook, lectures, or assignments. Faculty who regularly teach the class and the classes that follow (e.g., CSci 211, Computer Science III) participate in the evaluation, in the selection of textbooks, and in formulating a response appropriate to the assessment results.

How Data in the Course are Used to Assess Program Outcomes (unless adequately covered already in the assessment discussion under Criterion 4)

The standard exam administered in CSci 112 (see the previous item) is included in the curriculum–wide outcome assessment described in Chapters 2, 3, and 4 of the Self–Study.

Estimate Curriculum Category Content (Semester hours)

Area	Core	Advanced	Area	Core	Advanced
Algorithms	1		Software design		
Data structures	1		Concepts of programming languages	1	