COURSE DESCRIPTION

Dept., Number	CSci 555	Course Title	Functional Programming
Semester hours	3	Course Coordinator	H. Conrad Cunningham, Professor

Current Catalog Description

The principles and techniques of programming with functions. Purely functional programming languages; recursion; higher-order functions; reduction models; strictness; type systems; list operations; infinite data structures; program synthesis and transformation.

Textbooks

Simon Thompson. *Haskell: The Craft of Functional Programming, 2nd edition,* Addison Wesley, 1999.

H. Conrad Cunningham. *Notes on Functional Programming with Haskell.* Technical Report UMCIS-1995-01, partially revised 2007.

References

Class website: http://www.cs.olemiss.edu/~hcc/csci555/

Haskell language website: http://www.haskell.org

Course Outcomes

Upon successful completion of this course, the students:

1. know the concepts and terminology of the functional programming paradigm,
2. can solve problems by writing programs in the purely functional, lazily evaluated programming language Haskell,
3. are able to reason about and manipulate Haskell programs mathematically.

Relationship between Course Outcomes and Program Outcomes

This is a course taken primarily by beginning computer science graduate students; it is sometimes taken by undergraduate computer science students as an elective to enrich their programs. The course outcomes contribute to the program outcomes as follows: (1) to (a), (2) to (c), and (3) to (j).

Prerequisites by Topic

1. Basic discrete mathematics (Math 301)
2. Basic data structures and algorithms (CSci 112, 211)
3. Types in programming languages (CSci 112, 211)
4. Recursion (CSci 112, 211)

Major Topics Covered in the Course

1. Functional programming concepts. (3 hours)
2. Introduction to Haskell. First-order functions. Data types. (8 hours)
3. Higher-order functions. (8 hours)
4. List comprehensions. (1 hour)
5. User-defined data types. Trees. (2 hours)
6. Programming laws. Program synthesis techniques. (5 hours)
7. Monads and purely functional input/output. (2 hours)
8. Interpretation techniques. Program efficiency. (3 hours)
9. Type Classes. Modules. Abstract data types. (3 hours)
10. Infinite data structures. (1 hours)
11. Exams (3 hours)

Assessment Plan for the Course

This is an elective course offered approximately every two years primarily to computer science graduate students. An offering typically has 3 examinations and 6 programming and homework assignments. Outcome 1 is assessed by several exam questions, outcome 2 by 5 programming assignments and several exam questions, and outcome 3 by at least 1 homework assignment and several exam questions. The instructor evaluates the student performance informally and makes changes to the course content, organization, and pedagogy as appropriate for subsequent offerings of the course.

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

Estimate Curriculum Category Content (Semester hours)

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