COLLEGE of CHARLESTON

DEPARTMENT OF COMPUTER SCIENCE

Course Description

Computer Science Department, College of Charleston

Course Number: **CSCI 325**
Course Title: **Functional and Logic Programming**
Course Coordinator: **Jim Bowring**

Catalog Description

This course introduces various approaches to declarative (non-procedural) programming languages. Topics include mathematical functions and the lambda calculus; functional programming; sentential and predicate logic; and logic programming. *Prerequisites: CSCI 221 and MATH 207.*

Prerequisites by Topic

1. Object-oriented programming
2. Discrete structures

Major Topics Covered in the Course (Required Topics**)**

1. Overview of computational paradigms (1 hour)
2. Data and operations, functions, and recursion in Haskell (4 hours)
3. Lists in Haskell (3 hours)
4. Higher-order functions in Haskell (2 hours)
5. Lazy computing and functions with infinite output in Haskell (4 hours)
6. Introduction to the Lambda Calculus (2 hours)
7. Logic and mathematics in the Lambda Calculus (3 hours)
8. Truth-functional and Predicate logic (2 hours)
9. Introduction to Prolog (3 hours)
10. The declarative and procedural interpretations of Prolog (1 hour)
11. Lists in Prolog (3 hours)
12. Structured data in Prolog (3 hours)
13. Horn clauses, unification, backtracking, and resolution in Prolog (1 hour)
14. Church's Thesis, Turing machines, the Chomsky hierarchy (3 hours)
15. Tests (3 hours)

Course Narrative (optional)

(Describes what the instructor intends to achieve in the deliver of the course and in the learning of the students.)

Laboratory projects

1. Simple functions in Haskell. (1 week)
2. Recursive functions in Haskell. (2 weeks)
3. Higher-order functions in Haskell. (2 weeks)
4. Functions with infinite output in Haskell. (2 weeks)
5. Lambda-calculus exercises (pencil and paper). (2 weeks)
6. Relational predicates in Prolog. (1 week)
7. List-processing predicates in Prolog. (2 weeks)
8. Sorting in Prolog. (2 weeks)

Lasted Edited: 1/08/2013

COLLEGE of CHARLESTON

DEPARTMENT OF COMPUTER SCIENCE

Course Description

Computer Science Department, College of Charleston

Course Outcomes

Upon successful completion of the course, students will be able to:

Course Outcomes	Program Outcome Linkage
1. Explain imperative and non-imperative paradigms of computation.	aj
2. Explain the functional programming paradigm.	aj
3. Apply the Haskell language to programming problems.	abcj
4. Explain the lambda-calculus as a formalization of the functional paradigm.	a
5. Apply the lambda-calculus to logical and arithmetic problems.	a
6. Apply truth-functional and predicate logic to problems.	a
7. Explain the logic programming paradigm.	a
8. Apply the Prolog language to programming problems.	abcj
9. Explain the limitative theorems on logic and computation.	a

Oral and Written Communications

Every student is required to submit at least _0__written reports (not including exams, tests, quizzes, or commented programs) of typically __0_ pages and to make _0__oral presentations of typically _0__ minute's duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

Notes

e.g. special pedagogy, online component, etc.

Lasted Edited: 1/08/2013