Spring 2009

CS 776: Functional Programming

Krishnaprasad Thirunarayan
Wright State University - Main Campus, t.k.prasad@wright.edu

CS 776 Functional Programming

- **Instructor** : T. K. Prasad
- **Phone No.** : (937)-775-5109
- **Email** : t.k.prasad@wright.edu
- **Home Page** : http://www.cs.wright.edu/~tkprasad

- **Quarter** : Spring, 2009.
- **Class Hrs** : TTh, 4:10 - 5:25pm, **A330 Creative Arts**
- **Office Hrs** : TTh, 3-4pm, 395 Joshi Center (or by appointment)

Course Description

This course will discuss important concepts of functional programming such as recursive definitions, higher-order functions, type inference, polymorphism, abstract data types, modules etc. The programming exercises will illustrate the utility of list-processing, pattern matching, abstraction of data/control, strong typing, and parameterized modules (functors). We also study the mathematical reasoning involved in the design of functional programs and techniques for proving properties about functions so defined.

The programming assignments will be coded in SML '97 (Standard ML of New Jersey). and Haskell.

Prerequisites

- CS 480/680 Comparative Languages.

Course Texts

1. ML for the working programmer (2nd Ed.), L. C. PAULSON, Cambridge University Press, 1996. ISBN 0-521-56543-X.
2. Elements of ML Programming (ML 97 Ed.), J. D. ULLMAN, Prentice Hall, 1998. ISBN 0-13-790387-1.
3. Introduction to Functional Programming in Haskell (2nd Ed.), R. S BIRD, International Series in Computer Science. Prentice Hall, 1998. ISBN: 0-13-484346-0

References

1. Programming in Standard ML , Robert Harper, Carnegie Mellon University, 2005.
2. A Gentle Introduction to Haskell , Paul Hudak, John Peterson, Joseph Fasel, 2000.
3. Why Functional Programming Matters, John Hughes.
4. FP Tutorial (ps) (pdf). Benjamin Goldberg. PLDI-94.
5. The semantic elegance of applicative languages, David Turner, Proceedings of the 1981 conference on Functional programming languages and computer architecture, pp. 85-92, 1981.
6. The Standard ML Basis Library, Edited by Emden R. Gansner, John H. Reppy, Cambridge University Press, 2004.

Course Load

The course load includes a mixture of homeworks and programming assignments worth 50 points, a midterm worth 20 points and a final worth 30 points. Exams are typically open book.

Grading

The letter grades will be assigned using the following scale: A[90-100], B[80-90), C[70-80), D[60-70), and F[0-60). However, I reserve the right to adjust the scale somewhat to utilize the gaps in the distribution.

Attendance Policy

All registered students are expected to attend all lectures. In case a student is absent from a lecture due to unavoidable circumstances, the student is still responsible for the material covered in the class, as it is typically available from the course web-page well in advance. Furthermore, the student is expected to find out about in-class announcements from their colleagues/instructor.

Class Schedule and Syllabus

No.	Topic	Readings
Class 1	Functional Programming Basics	Chap. 1 (LP) Chap. 1 (JU)
Class 2	Higher-order functions	Chap. 2, 5 (LP) Chap. 2, 5 (JU)
Class 3	Type inference	Chap. 2, 3 (LP)
Class 4	Polymorphic Type System	
	(Fixed Points)	
Class 5	Programming with lists: Pattern matching	Chap. 3 (LP) Chap. 3-5 (JU)
Class 6	SML-97 Specifics	
Class 7	Introduction to Haskell	
Class 8	Fold operations : `foldr, foldl`	Chap. 5 (LP) Chap. 5 (JU)
Class 9	**Midterm Exam (April 28)**	
Class 10	Types : Concrete and Abstract	Chap 6 (JU)
Class 11	Examples	Chap 7 (LP)
Class 12	Modules	Chap 7 (LP) Chap. 8 (JU)
Class 13	(cont'd)	
Class 14	Recursion and Induction	Chap. 6 (LP)
Class 15	Examples (Combinatorial Functions)	
Class 16	Records; Exceptions; References;	Chap 8 (LP) Chap. 7 (JU)
Class 17	Efficiency; Streams;	Chap. 5 (LP)
Class 18	Introduction to Haskell	
Class 19	(Adv. Haskell) (haskell.pdf)	
Class 20	Lambda Calculus	
	Final Exam (June 9, 5:45-7:45pm)	

Assignments (`Spring 2009`)

- Assignment 1 (asg1.html)
- Assignment 2 (asg2.html)

Examinations (`Spring 2007`)