Course Development

Position in Curricula

The course CSci 555, Functional Programming, satisfies the requirements for a programming intensive course in the MS and PhD programs in computer science. It can also be used as an elective in the undergraduate curriculum.

History

I created CSci 555 at the University of Mississippi and first taught it during the Spring 1991 semester.

I adopted the first edition of Bird and Wadler [Bird 1988] as the initial textbook for the course. I thank Richard Bird and Philip Wadler for writing this excellent textbook. I thank Jeremy Gibbons for suggesting that book in a response to an inquiry I posted to a Usenet news group in Summer 1990.

I also used Wentworth’s RUFL (Rhodes University Functional Language) interpreter and his tutorial [Wentworth 1990] in the first two offerings of the course. I thank Peter Wentworth for sending me (unsolicited, in response to my Usenet

[Copyright (C) 2019, H. Conrad Cunningham
Professor of Computer and Information Science
University of Mississippi
211 Weir Hall
P.O. Box 1848
University, MS 38677
(662) 915-5358]
his interpreter and tutorial on a floppy disk through snail mail from the then-sanctioned South Africa.

My approach was also shaped by my research on formal methods and my previous teaching on that topic. I created the course Program Semantics and Derivation (CSci 550) and first taught it in Spring 1990. I followed that with the course Theory of Concurrent Programming (Engr 664), which I first taught in Fall 1990. I thank my dissertation advisor Gruia-Catalin Roman for developing my interests in formal methods, Jan Tijmen Udding for teaching a graduate course on program derivation that piqued my interests, and the other researchers or authors who have influenced my thinking: Edsger Dijkstra, Tony Hoare, David Gries, Mani Chandy, Jayadev Misra, Edward Cohen, and many others.


For the third offering of CSci 555 in Fall 1993, I switched the course to use the Gofer interpreter for the Haskell language. I thank the international committee of researchers, including Simon Peyton Jones, Paul Hudak, Philip Wadler, and others, who have developed and sustained Haskell since the late 1980s. I also thank Mark Jones for developing the lightweight Gofer interpreter and making it and its successor HUGS widely available.

Because of the need for a tutorial like Wentworth’s and an unexpected delay in getting copies of the Bird and Wadler textbook [Bird 1988] from Prentice Hall that semester, I began writing, on an emergency basis, what evolved into my Notes on Functional Programming with Haskell [Cunningham 2014]. Some parts of the Notes were based on my handwritten class notes from the the 1991 and 1992 offerings of the course. Many pages of the Notes were written “just-in-time” in late-night sessions before I taught them the next day. I thank Prentice Hall (now Pearson) for its delay in shipping books across the “big pond”, my wife Diana Cunningham for tolerating my disruptive schedule, and my Fall 1993 students for not complaining too vigorously about a quite raw set of class notes.

I continued to develop the Notes for the Fall 1994 and Fall 1995 offerings of the course. In early 1996, I created a relatively stable version of the Notes that I continued to use in subsequent offerings of CSci 555. I thank my students and others who pointed out typos and suggested improvements during the 1993-1996 period.

I thank David Gries for encouraging me to expand the Notes into a textbook. I am doing that, albeit over 20 years later than Gries intended.

For the Spring 1997 offering of CSci 555, I started using the new HUGS interpreter (which replaced Gofer) and the 1st edition of Thompson’s textbook [Thompson
1996] (now its 3rd edition). I thank Simon Thompson for writing his excellent, comprehensive introductory textbook on Haskell programming and Mark Jones for developing HUGS.

Over the next 17 years, I corrected a few errors but otherwise the Notes were stable. However, I did create supplementary notes for CSci 555 and related courses. These drew on the works of Abelson and Sussman [Abelson 1996], Thompson [Thompson 1996], Parnas [Parnas 1972] [Britton 1981], and others.

In Fall 2014, I taught CSci 450 (Organization of Programming Languages) using Haskell and used about half of the Haskell functional programming notes. I revised the Haskell Notes to use Haskell 2010 and the GHC compiler.

In Summer 2016, I began developing the Haskell-based textbook now titled Exploring Languages with Interpreters and Functional Programming (ELIFP) [Cunningham 2018] for CSci 450. This textbook absorbed most of the Haskell material (although not all is used in 450) and expanded it to include more discussion of programming paradigms, software testing, modularity, language processing, and interpreters. I first used ELIFP in CSci 450 in Fall 2017.

As a result of using Haskell in the Fall 2014 offering of CSci 450, I revised the Spring 2016 offering of CSci 555 to use Scala instead of Haskell. I chose the “red book” [Chiusano 2015] as the textbook.

In Spring 2017, I used CSci 556 (Multiparadigm Programming) to revisit the functional programming language Haskell and explore some additional areas in preparation for using Haskell in the fall 2017 offering of CSci 450. In particular, I used that opportunity to continue the work on the ELIFP textbook.

For the Spring 2017 offering of CSci 555, I am again using Scala, but I do not plan to use the “red book”.

References


