CS 790: Functional Programming

Tian Zhao

Spring 2020

Course Description: This course is an introduction to the concepts of functional programming using Haskell as the instruction language. The goal of this course is to provide foundation for students who are interested in software development and research using functional programming concepts. While pure functional languages such as Haskell, LISP, and ML are not popular choice for programmers, functional programming concepts are found in many mainstream languages such as Scala, C#, Python, and JavaScript, as well as newer languages such as Clojure, Go, Julia, Kotlin, TypeScript, and PureScript. Students will be able to recognize similar concepts in these languages and use them effectively. A secondary goal of the course is to learn how to use functional concepts to design embedded domain specific languages for specialized tasks such as machine learning and IoT applications.

Time and Location:	**Section 002**	1/21 – 5/7	TR 5:30 PM – 6:45 PM	EMS E220

Prerequisite: CompSci 431 or equivalent or graduate standing

Level: Graduate.

Instructor: Tian Zhao
Office: EMS 1145, Telephone: 414 229 5682, Email: tzhao@uwm.edu,
Office hour: TR: 1PM – 2PM or by appointment.

Textbook:

1. Learn You a Haskell for Great Good! (online free) `http://learnyouahaskell.com/chapters`
2. Real World Haskell. (online free) `http://book.realworldhaskell.org/read/`

Class Web Pages: We will use the canvas website located at `https://uwm.edu/canvas`
Assignments, announcements, and class notes will be posted on the course web page.

Grading: homeworks: 60%
midterm: 20%
final: 20%

Homework Assignments: Number of homeworks and their due dates are to be announced. Homeworks are mostly programming assignments using Haskell.

Midterm and Final Exams:

Section 002			
Midterm	5:30 pm – 6:45 pm	Thursday, 03/05	EMS 220
Final exam	5:30 pm – 7:30 pm	Tuesday, 05/12	EMS 220

Course Schedules: The topics covered in this course are listed in the following tables. Most of the lectures will be related to chapters in the textbook but some lectures may use research papers as source material. The following schedule is tentative and may be subject to changes.

Date	Subject
week 1 01/21	course introduction/Haskell preliminaries
week 2	Constants, variables, tuples, lists, and functions
week 3	Types and builtin type class
week 4	Pattern matching and recursion
week 5	Higher order functions and modules
week 6	Algebraic data types and type class definition
week 7 03/5	I/O **midterm**, Thursday
week 8	Functors, Applicatives, and Monoids
week 9	Monads
week 10	Parser Derivation
week 11	Monad Transformers
week 12	Program Verification
week 13	Embedded Domain Specific Language
week 14	Type-based modeling
week 15	Software Transactional Memory
Final	take home exam

Accessibility: If you will need accommodations in order to meet any of the requirements of this course, please contact me as soon as possible.

Academic Misconduct: Student academic misconduct procedures are specified in Chapter UWS 14 and the UWM implementation provisions (Faculty Document 1686).

Academic misconduct is an act in which a student seeks to claim credit for the work or efforts of another without authorization or citation, uses unauthorized materials or fabricated data in any academic exercise, forges or falsifies academic documents or records, intentionally impedes or damages the academic work of others, engages in conduct aimed at making false representation of a student's academic performance, or assists other students in any of these acts.

Prohibited conduct includes cheating on an examination; collaborating with others in work to be presented, contrary to the stated rules of the course; submitting a paper or assignment as one's own work when a part or all of the paper or assignment is the work of another; submitting a paper or assignment that contains ideas or research of others without appropriately identifying the sources of those ideas; stealing examinations or course materials; submitting, if contrary to the rules of a course, work previously presented in another course; tampering with the laboratory experiment or computer program of another student; knowingly and intentionally assisting another student in any of the above, including assistance in an arrangement whereby any work, classroom performance, examination or other activity is submitted or performed by a person other than the student under whose name the work is submitted or performed.