

CSci 555: Functional Programming

Spring Semester 2019

Syllabus

H. Conrad Cunningham

23 January 2019

Course Overview

Catalog Description

CSci 555, Functional Programming: 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.

Prerequisites

Either CSci 211 and Math 301 or graduate standing.

I expect students to have previous programming experience and knowledge similar to completion of the CSci 111-112-211 sequence (which uses object-oriented programming in Java). I also expect students to have a basic understanding of discrete mathematics (e.g. sets, relations, functions, induction, and simple algebraic concepts) similar to completion of Math 301.

Position in Curricula

CSci 555, Functional Programming, is an elective course for the BA, BSCS, MS, and PhD programs in computer science. It can satisfy the programming intensive requirement of the MS and PhD emphases in computer science.

Course Goals

1. This course enables graduate and upper-division undergraduate students to develop nontrivial computer programs within the functional programming paradigm using the chosen language. The programs should correctly meet

their requirements, be algorithmically efficient, be robust and flexible, and be developed using appropriate design and programming methods.

2. The students should learn to think and to solve problems using the concepts of functional programming and be able to apply those concepts in languages other than the chosen language for the course.

For the Spring 2019 semester, this course will use the polymorphically statically typed, hybrid object-functional programming language Scala.

Course Student Outcomes

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

1. describe the concepts of the functional programming paradigm (BSCS outcome 6)
2. recognize syntactically and semantically correct functional programs written in the chosen language (BSCS outcome 6)
3. analyze complex computing problems and identify the requirements for functional programming solutions (BSCS outcome 1)
4. develop (i.e. design, implement, execute, and test) functional programs written in the chosen language that satisfy their requirements (BSCS outcomes 2, 6)
5. evaluate alternative functional programs to determine which are better according to selected criteria (BSCS outcomes 2, 6)
6. relate the concepts and methods for functional programming in the chosen language to their previous programming knowledge and experiences (BSCS outcomes 2, 6)
7. be more confident as computer programmers using a variety of languages and techniques (BSCS outcomes 1, 2, 6)
8. appreciate the ability to exploit the abstraction and mathematical properties of a purely functional programs and languages (BSCS outcome 6)
9. study and learn a programming paradigm and language with which they previously had little knowledge (BSCS outcomes 2, 6)
10. use and construct libraries that abstract recurring program aspects using features the chosen language (BSCS outcomes 2, 6)

Students taking the course for graduate credit must demonstrate the ability to analyze scientific literature on functional programming or related topics.

Spring 2018 Section Details

Time and Place

11:00-11:50 a.m. Monday-Wednesday-Friday; Hume Hall 110

Instructor

Dr. H. Conrad Cunningham	Professor, Computer and Information Science
Office:	211 Weir Hall
Telephone:	(662) 915-5358
Email:	hcc@cs.olemiss.edu
Web:	http://www.cs.olemiss.edu/~hcc
Office hours:	Monday-Wednesday 1:00 - 3:00 p.m. (or by arrangement)

Teaching Assistant

None for the Spring 2019 semester

Communication Policy

Students may contact the instructor by telephone or email or meet him during his office hours. He will attempt to respond to email and telephone messages within 24 hours during the work week. Emails or telephone calls arriving outside the 8:00 a.m. to 5:00 p.m., Monday-Friday workday may be deferred until the next workday.

When needed, the instructor uses Blackboard's announcement feature to send messages to the entire class. This feature sends emails to the class members and archives them on Blackboard's course announcement page.

Methods of Instruction

The instructional methods used in this course include readings from the textbooks and other course material; classroom lectures, discussions, and activities; programming and other assignments, quizzes, and a semester project.

Course Materials

I will maintain a CSci 555 course website at:

<http://www.cs.olemiss.edu/~hcc/csci555/csci555.html>

Most of the course materials will be linked to the Lecture Notes page. Any restricted access materials will be posted to the course's Blackboard site.

There is no required textbook, but I submitted the following as a recommended reference book on Scala:

- Martin Odersky, Lex Spoon, and Bill Venner. *Programming in Scala*, 3rd edition, Artima Inc, 2016.

Odersky's research group at EPFL was the original developers of Scala.

Other materials I have used in the past for my Scala-based courses:

- *Notes on Scala for Java Programmers* [HTML] [PDF].

I adapted this work from a tutorial on the Scala language website.

- Martin Odersky. *Scala by Example*, EPFL, 2015. Available at <https://www.scala-lang.org/docu/files/ScalaByExample.pdf>.
- Paul Chiusano and Runar Bjarnason. *Functional Programming in Scala*, Manning Publications, 2015.

This is sometimes called the Red Book. I used this book for CSci 555 in Spring 2016 and have lecture notes for chapters 3-5 on the course's Lecture Notes page.

For some material, we will use selected chapters or sections of:

- Dave Gurnell and Noel Welsh. *Creative Scala*, Underscore Consulting LLP, 2018. Available at <https://www.creativescala.org/creative-scala.html>.
- Noel Welsh and Dave Gurnell. *Essential Scala*, Underscore Consulting LLP, 2017. Available at <https://books.underscore.io/essential-scala/essential-scala.html>.
- Noel Welsh and Dave Gurnell. *Scala with Cats*, Underscore Consulting LLP, 2017. Available at <https://underscore.io/books/scala-with-cats/>.
- Dave Gurnell. *The Type Astronaut's Guide to Shapeless*, Underscore Consulting LLP, 2017. Available at <https://underscore.io/books/shapeless-guide/>.
- Tutorials and documentation for libraries and frameworks we use.
- Other items to be determined.

Hardware

The course does not require any special hardware. Students may use the Department's Adler lab facilities or servers or use their own desktop or laptop systems to complete the programming assignments.

Software

To complete the course, students need the following software available on the computer at which they will do their work:

- Scala 2.12 (which requires Java 8)

Note: Scala 2.13 may be released during the semester. This may introduce some incompatibilities (especially in some of the libraries), so we may not want to upgrade immediately,

- Text editor(s) or development environment(s) for editing Scala source code (Eclipse, IntelliJ, Emacs, Vim, Atom, etc.)
- Additional Scala tools and libraries to be determined (e.g. `sbt` build tool, Cats libraries, Akka actors, Spark framework)
- A browser with plug-ins compatible with Blackboard

Students can run the Blackboard Browser Check to verify the configuration of the browser for normal use of Blackboard.

- A browser that supports MathML to display the HTML version of the instructor's course notes

The current version of Mozilla Firefox should work appropriately for both MathML and Blackboard.

Technical Support

Contact the instructor concerning any issues related to installation and use of the language software or if access is needed to Department of Computer and Information Science computing facilities.

Course Topics

The course is expected to cover many of the following topics. Of course, this list is subject to refinement as the semester progresses.

- Functional programming paradigm
- Introduction to Scala (for Java programmers): functions, recursion, objects, classes, case classes, pattern matching, traits, generics
- Functional programming concepts: referential transparency, recursion, first-class functions, higher-order functions, polymorphism, programming patterns, type-driven design
- Functional data structures, error handling
- Type classes, monoids, functors, monads, Cats library
- Concurrency: futures, actors, parallel streams
- Other advanced topics

Course Expectations and Attendance Policy

The instructor expects each student to attend class and participate during each scheduled class period and to complete the required out-of-class assignments by the stated deadlines.

All students are expected to study the relevant portions of the course materials in conjunction with the class discussions (i.e. before coming to class). Explicit reading assignments will not always be given. If in doubt on what you need to read, please ask the instructor.

Grading

Component	Weight
Assignments	45
Quizzes	30
Semester Project	15
Class Participation	10

Assignments and Projects

- Each student is expected to complete all required assignments.
- The assignments for students taking the course for undergraduate credit will be modified appropriately.
- Several programming assignments will be assigned during the semester. We will primarily use the Scala 2.12 language.
- Some of the assignments may require study of scientific papers or technical documentation and submission of written documents or presentation of material to the class.
- Unless otherwise stated in the assignment description, an assignment is to be carried out by each individual student without inappropriate collaboration with others. See the section on *Academic Integrity*.
- Most assignments will be submitted electronically using the Blackboard system.
- In preparing and submitting homework assignments make sure that:
 - your name, the course number or name, the assignment identifier, and individual exercises are clearly indicated in the content of the file or on the paper. (If it is a group assignment, give the group identifier and the names of all members.)
 - for any handwritten portions, you write legibly on only one side of the paper in a black or blue pen or dark pencil. Do NOT use red or green ink!
 - for any assignments submitted on paper, your pages are stapled together in the upper left corner when viewed from the front.

- if you reference outside textbooks, reference books, articles, websites, etc., or discuss an assignment with individuals inside or outside the class, you must document these by including appropriate citations or acknowledgements in your submitted work (e.g. a comment in the header of the primary source file of a program).
- All students are expected to complete their homework assignments by their due dates. If an assignment is submitted late, a penalty of 10 percent of that assignment’s grade will be assessed for each class day it is late. A homework paper will not be accepted after graded papers have been returned, after a solution has been distributed, or after the final examination.
- If an assignment is marked “optional”, it can replace an earlier assignment. However, all students are encouraged to work on these assignments; these optional activities can help the student master the concepts and skills and prepare for the proctored exams.
- The class project will be announced later. It likely will include a combination of study of scientific literature and technical documentation, programming, writing, and presentation.

Examinations and Quizzes

- There will be no regular, in-class examinations.
- There will be several quizzes throughout the semester.
- Quizzes may be online or in-class; the latter will be designed to only take a portion of the class time.
- Quizzes may be either announced or unannounced.
- There will be at least one drop quiz. There will be no makeup quizzes.
- There may be an optional, *practical* exam during the final exam period. We will discuss this later in the semester.

University of Mississippi Policies

Disability Access and Inclusion

The University of Mississippi is committed to the creation of inclusive learning environments for all students. If there are aspects of the instruction or design of this course that result in barriers to your full inclusion and participation, or to accurate assessment of your achievement, please contact the course instructor as soon as possible. Barriers may include, but are not necessarily limited to, timed exams and in-class assignments, difficulty with the acquisition of lecture content, inaccessible web content, and the use of non-captioned or non-transcribed video and audio files.

If you are approved through Student Disability Services (SDS), you must log in to your *Rebel Access* portal at <https://sds.olemiss.edu> to request approved accommodations. If you are NOT approved through SDS, you must contact Student Disability Services at 662-915-7128 so the office can:

1. determine your eligibility for accommodations,
2. disseminate to your instructors a Faculty Notification Letter,
3. facilitate the removal of barriers, and
4. ensure you have equal access to the same opportunities for success that are available to all students.

Any student requesting accommodation under this policy should present the instructor with the required documentation early in the semester and make arrangements in advance for each examination or assignment.

Copyrighted Materials

Materials used in connection with this course may be subject to copyright protection under Title 17 of the United States Code. Under certain Fair Use circumstances specified by law, copies may be made for private study, scholarship, or research.

Students should not share electronic copies of copyrighted materials with unauthorized users. Violations of copyright laws could subject individuals to federal and state civil penalties and criminal liability as well as disciplinary action under University policies.

Appropriate Use of Information Technology

The Information Technology (IT) Appropriate Use Policy sets forth the privileges of and restrictions on students, faculty, staff, and other users with respect to the computing and telecommunications systems offered by the University of Mississippi (UM). This policy is designed to protect the University community from illegal or damaging actions by individuals, either knowingly or unknowingly. Inappropriate use exposes the University to risks, including virus attacks, compromise of network systems and services, and legal issues. This policy directly addresses copyright issues related to illegal downloads and peer-to-peer file sharing.

For questions about the Appropriate Use Policy, send an email to aup@olemiss.edu.

Academic Integrity

The University of Mississippi seeks to support and sustain a safe and scholarly community of learning dedicated to nurturing excellence inside and outside the classroom. The University expects all students to familiarize themselves with the values and standards reflected in its policies and to honor those values and

standards in their conduct as individuals and groups. These policies are outlined in the *M Book*. The University Policy Directory gives a complete listing of the policies.

As a student in Computer Science Senior Project, you are expected to conduct yourself in a professional and ethical manner according to the policies, procedures, and expectations of the Department of Computer and Information Science, School of Engineering and College of Liberal Arts, University of Mississippi, and discipline of computer science.

This course will strictly follow the University's academic discipline policies and procedures. Academic misconduct may result in penalties ranging from a grade reduction on an assignment to expulsion from the course, depending on the severity of the violation.

Verification of Student Attendance

The University must abide by Federal guidelines to verify the participation of students. For all course types, including thesis, internships, labs, online courses, etc., the instructor must verify your participation based on some type of participation. In this course, the instructor will verify the physical attendance of each student and report it during the first two weeks and otherwise as required.

Student Privacy Policy

The University of Mississippi protects the privacy of all students, including online and distance learning students, through adherence to the Family Educational Rights and Privacy Act of 1974 (FERPA) through compliance with other institutional policies and procedures governing the management and security of protected information of faculty, staff, and students, and by outlining the expectations of privacy for the university community as regards to electronic information. See the *Student Information and Privacy Policy* for more information.