CSci 658-01: Software Language Engineering
Spring 2018 Syllabus

H. Conrad Cunningham

22 January 2018 (revised 13 April 2018)

Professor of Computer and Information Science
University of Mississippi
211 Weir Hall
P.O. Box 1848
University, MS 38677
(662) 915-5358

Course Overview

Introduction

In this course, we study the design, implementation, and use of both external and internal (or embedded) domain-specific languages (DSLs).

A DSL is a simple language normally with a syntax and semantics focused on a specific application area, perhaps meant to be used by experts on that application area who are not necessarily expert programmers. An external DSL consists of a language separate from the DSL processor's host language. An internal DSL consists of the restricted usage of the host language itself to create the specialized language syntax and semantics.

Domain-specific languages and language-oriented programming have been topics of considerable interest in recent years, both in industrial applications and in academic research. For example, the popular Ruby on Rails web framework consists of several "internal" DSLs in the language Ruby, and Microsoft supports DSL development and use as a part of its Visual Studio toolset. On the Java platform, JMock is an example of an internal DSL to support unit testing.

Most software engineers will use DSLs in the future as a part of their work. Many will be called upon to design them. This course seeks to address both needs, with an emphasis on the latter.

In this course, we will also delve into the metaprogramming features of languages, particularly for internal DSLs.

Catalog Description

Course Description from Catalog

Study of topics on the design, implementation, use, and evolution of artificial languages for the engineering of software. Languages of interest include general-purpose programming languages, domain-specific languages, and modeling languages as well as application programming interfaces and collections of design patterns that implicitly define languages.

Prerequisites

Graduate standing in computer science

Position in Curricula

This is a 600-level, programming-intensive course intended primarily for MS and PhD students in computer science.

Course Goal

The primary goal of this course is to enhance the students' knowledge of and experience with language-oriented programming techniques, in particular, with the design, implementation, and use of domain-specific languages (DSLs).

A secondary goal is to explore advanced features of one or more programming languages useful in implementation of internal and external DSLs.

Course Student Outcomes

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

1. describe the fundamental concepts and methods of domain-specific languages (DSLs), including both external and internal DSLs
2. analyze problems and apply DSL concepts and methods to design DSL solutions, if appropriate to the problem and environment
3. apply external DSL implementation methods and tools to develop text-based DSLs using advanced features of the selected implementation language(s)

4. apply internal DSL implementation methods and libraries to develop internal DSLs using advanced features of the selected host language(s)
5. evaluate alternative DSL designs and implementations to determine which are more appropriate according to the selected criteria
6. relate the DSL-related programming and metaprogramming concepts and methods to their previous programming knowledge and experiences
7. appreciate the elegance and usefulness of approaching software development as a language design and implementation activity
8. be confident in their abilities to study unfamiliar scholarly papers, technical documentation, and program designs and implementations and apply what they learn to new situations

Spring 2018 Section Details

Time and Place

10:00 - 10:50 a.m. Monday-Wednesday-Friday; Weir Hall 106

Instructor

Dr. H. Conrad Cunningham, Professor, Computer and Information Science
Office: 211 Weir Hall
Telephone: (662) 915-5358
Email: hcc@cs.olemiss.edu
Web: Professor Cunningham's Homepage (http://www.cs.olemiss.edu/~hcc)
Office hours: 11:00 - 12:00 MWF; 1:00 - 2:00 MW; by arrangement

Teaching Assistant

None

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.

If there is a Teaching Assistant assigned, students may also meet with the Teaching Assistant during his or her office hours or by email.

The instructor and teaching assistant will use the features of the Blackboard Learn system (announcements and email) to communicate urgent information to students outside of class.

Methods of Instruction

The instructional methods used in this course include readings from professional papers, textbooks, the instructor's lecture notes and case studies, and other course materials; classroom lectures and discussions; assignments (DSL analysis, design, and implementation exercises, other programming and design exercises, and quizzes); a term project, and exams.

Note: Near mid-semester, the six students and I agreed to eliminate the planned exams and focus the course around the assignments, an in-class exercise, and the semester project.

Course Materials

The Spring 2018 offering will primarily use the Python 3 programming language.

Note: I subsequently relaxed the language usage, using other languages in my presentations and allowing students to use other languages in their programming projects. However, all students actually used Python 3 and I used Python 3 for most new examples and notes I created.

Course Websites

The instructor will maintain a CSci 658 course website on the Department's server at http://www.cs.olemiss.edu/~hcc/csci658/csci658.html. Most course materials will be linked to the Lecture Notes page.

The instructor will post any materials requiring restricted access on the course's UM Blackboard site.

Required Textbooks

None

Recommended Textbooks

- Martin Fowler. *Domain Specific Languages*, Addison Wesley, 2011.
- Luciano Ramalho. *Fluent Python*, O'Reilly. 2015.

Other Readings

- Journal and conference papers and book chapters related to the topics discussed in the course
- Web-hosted materials (documentation, tutorials, blogs, etc.) on course-related topics (DSLs, Python 3, useful tools, etc.)

Hardware

The course does not require any special hardware. Students may use the Department's Adler lab facilities, servers, and office systems or 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:

- Python 3.6+
- Various Python 3 library modules (to be determined as the semester progresses)

  Note: The only non-standard Python library used was the parsing combinator library Parsita, which was used in one DSL example. I also used the GraphViz (dot) tool for another DSL example.
- Tools for developing Python 3 programs and DSLs
- 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 or teaching assistant 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

This is an experimental course offering, so the content will be revised and elaborated as the semester progresses.

1. Python 3 programming fundamentals.
2. Selected advanced Python 3 features (e.g., modules, objects, meta-object protocol, generators, higher-order functions, decorators, annotations).
3. Domain-specific language (DSL) concepts.
4. Analysis, design, and implementation methods for DSLs.
5. External DSL design patterns. Relevant features of Python 3 and associated libraries and tools.
6. Internal DSL design patterns. Relevant features of Python 3 and associated libraries and tools.

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.

Course Components

The original course structure used the following weightingL

Component	Weight
Class Participation and Attendance	10%
Assignments and Term Project	45%
Examinations	45%

At mid-semester, the students and I agreed to use the following weighting:

Component	Weight
Class Participation and Attendance	20%
Assignments 1-3	48%
Term Project	32%

Component	Weight
Examinations	0%

Class Participation

- All students should study the relevant portions of the course materials in conjunction with the class discussions and activities (i.e., before coming to class). The instructor will not always give explicit reading assignments, but the Lecture Notes web page should give the topics planned for the next class. If in doubt on what you need to read, please ask the instructor.
- The instructor expects each student to participate in class discussions and activities.
- The Department expects graduate students (especially those with assistantships) to attend most Department seminars.
- To encourage class participation and attendance, the instructor will check the roll and may give written or oral quizzes or other short reports.

Assignments and Projects

- There will be several assignments. One of these will be identified as the term project, which will bear about one-half of the credit for the assignment portion of the semester grade. You likely will be asked to design a DSL and provide implementations.

  Note: The revised course structure differed from the above.
- Each student is expected to complete all required assignments.
- Assignments may involve analysis of problems and the design, implementation, and testing of DSLs or the learning of supporting concepts, methods, and tools. An assignment submission may include a mix of design documents and programs.
- Most of the assignments will involve programming. We will primarily use the Python 3 language, but we may use other languages as appropriate.
- Unless otherwise stated in the assignment description, each student is expected to carry out each assignment 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 each 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, staple your pages 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.

Examinations

- There will be two examinations: one near mid-term and one near the end of the semester. The final examination period published by the Registar's office (Friday, 11 May, 8:00 a.m.) may be used for the second exam, term project presentations, or other class activities.

  Note: The revised course structure eliminated the exams and placed greater weight on the assignments.

- Please do not ask to take the final examination earlier than the time set for the entire class.

- Each exam will cover all topics studied to that point. The final exam is comprehensive over all material covered during the semester.

- Each exam may consist of some combination of in-class and take-home components.

- If you cannot take an examination at the scheduled time because of an illness or other special circumstances, please notify the instructor in advance.

Without advance notification, it may not be possible to give a make-up examination.

Optional Activities

The instructor may suggest various optional resources and activities. These activities are not graded, but they can help a student broaden and deepen his or her understanding of the DSL and advanced programming language concepts.

Grading

- We will attempt to grade submitted assignments within approximately one week.
- The ranges for the letter grades are 90-100% for A, 80-89% for B, 70-79% for C, 60-69% for D, and below 60% for F.
- +/- grades will be used near the boundaries at the discretion of the instructor. The actual assignment of +/- grades depends upon the overall distribution of grades for the class.

University of Mississippi Policies

Students with Disabilities

University policy provides for reasonable accommodations to be made for students with verified disabilities on an individualized and flexible basis as specified under Section 504 of the Rehabilitation Act of 1973 and the Americans with Disabilities Act of 1990 (ADA).

Students with disabilities who believe they may benefit from classroom or other accommodations should contact the *Office of Student Disability Services* for information at 234 Martindale, 662-915-7128 (Phone), 662-915-7907 (TTY Phone), or sds@olemiss.edu (Email).

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 is dedicated to supporting and sustaining a safe and scholarly community of learning dedicated to nurturing excellence inside and outside of the classroom. Each student has a duty to become familiar with University values and standards reflected in University policies, and each student has a duty to honor University values and standards reflected in University policies. These policies are outlined in the *M Book*. For a complete listing of policies, please visit the University Policy Directory.

As a student in CSci 658, 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, Graduate School, University of Mississippi, and discipline of computer science.

The University's academic discipline procedure will be followed in this course. Violations of academic integrity may result in anything from failure 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.