Course Overview

Motivation

Language researcher and author Timothy Budd defines a programming paradigm as “a way of conceptualizing what it means to perform computation, of structuring and organizing how tasks are to be carried out on a computer” [Budd 1995, p. 3].

Budd reports, “Research results from the psychology of programming indicate that expertise in programming is far more strongly related to the number of different programming styles understood by an individual than it is the number of years’ experience in programming” [Budd 1995, p. viii]. He also states that the “goal of multiparadigm computing is to provide … a number of different problem-solving styles” so that a programmer can “select a solution technique that best matches the characteristics of the problem to be solved” [Budd 1995, p. 6]. A multiparadigm programming language is then a language that provides convenient support for two or more programming paradigms.

This course seeks to prepare students for roles as software designers and programmers in the future workplace. In the future, most expert programmers will need to be familiar with programming in multiple paradigms. Most languages are evolving to include features drawn from multiple paradigms—imperative, functional, object-oriented, concurrency, etc. New multiparadigm languages are emerging.

This course also seeks to prepare graduate students to conduct research on related software engineering and programming language topics.

Catalog Description

CSci 556, Multiparadigm Programming: Principles and practices of software design and programming using languages that feature explicit and convenient support for multiple programming paradigms (e.g., imperative, object-oriented, and functional).

Prerequisites

Either CSci 211 and 223 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 will teach CSci 556 as an intermediate/advanced course in Python programming, so students who do not have previous Python experience will be expected to learn Python fundamentals on their own.

Position in Curricula

CSci 556, Multiparadigm 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 Goal

The primary goal of this course is to deepen the students’ knowledge and experience with various programming paradigms and methods.

Course Student Outcomes

Upon successful completion of this course, students:

1. should know and understand the fundamental concepts, techniques, and terminology of several programming paradigms (e.g. imperative, functional, logic, modular, object-oriented, concurrent, reactive, parallel, and language-oriented paradigms)

2. can analyze problems and apply these programming concepts and techniques to develop programs to solve the problems

3. can evaluate alternative programming solutions to subproblems and integrate the solutions to produce effective multiparadigm programs in the chosen language
In particular, the Spring 2018 offering will use Python 3 and focus on the (a) imperative, (b) modular, (c) object-oriented, and (d) functional (using iterators, generators, and comprehensions) paradigms with optional coverage of metaprogramming and concurrent or coroutine programming.

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

Spring 2018 Section Details

Time and Place

2:00 - 2:50 p.m. Monday-Wednesday-Friday; Weir Hall 235

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: 10:15 - 11:30 a.m. MWF (or 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.

Methods of Instruction

The instructional methods used in this course include readings from the textbooks and other course material; classroom lectures, discussions, and activities; assignments (programming projects, written exercises, and quizzes), and in-class examinations.

3
Course Materials

I will maintain a CSci 556 course website at http://www.cs.olemiss.edu/~hcc/csci556/csci556.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:


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

- H. Conrad Cunningham. Exploring Languages with Interpreters and Functional Programming, draft textbook, 2018. [HTML] [PDF]

For some topics, I have notes, slides, and examples developed originally for other languages (e.g. Java, Scala, Elixir, Haskell, Lua, Ruby) or other courses (e.g. Software Language Engineering, Software Families, Software Architecture, Functional Programming) that we may use with or without being adapted to Python 3.

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:

- Python 3.7
- Additional Python 3 packages to be determined
- Text editor(s) or development environment(s) for editing Python 3 source code
- 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 most of the following topics. Of course, this list is subject to refinement as the semester progresses.

• Programming paradigms
• Python 3 types
• Python 3 imperative programming fundamentals
• Data abstraction and modular programming
• Object-based paradigms and object-oriented programming
• Functional programming
• (tentative) Metaprogramming
• (tentative) Software testing
• (tentative) Concurrent or coroutine programming
• (tentative) Other topics of interest to instructor and students

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.

Assignments

• All students are expected to study the relevant portions of the course materials in conjunction with our 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.

• 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 Python 3.7 language. There may also be written assignments or short quizzes. (The instructor anticipates approximately six major assignments, but may add smaller assignments to replace or augment the major assignments as appropriate.)

• 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.

Examinations

• There will be three examinations—two mid-term exams and a final exam.
• The lowest grade of the three will be dropped and each of the other two grades will contribute 25 percent toward the course grade.

• The mid-term examinations will be given in late September and in mid-November. The final examination will be given during the final examination period published by the Registrar’s office (Monday, 9 Dec, 4:00 p.m.).

• 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 a 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 programming language concepts.

Grading

• 50% of the final grade comes from the grades on the examinations. The remaining 50% comes from grades on the assignments (programming project, written exercises, and quizzes).

• The general 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.

• We will attempt to grade submitted assignments within approximately one week.
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.

Copyrughted 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 556, 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.