INLS 560 – Programming for Information Professionals

Instructor:

Jason Carter   Email: carterjl@cs.unc.edu

Lectures:

Room: Manning 117   Time: Friday – 9:00 am - 11:45 am

Note: Bring your laptops to class everyday. You will use them for programming exercises.

Official Course Description

An introduction to computer programming focusing on language fundamentals and programming techniques for library and information science applications. Emphasizes problem-solving through the development of practical applications that include text processing, file handling, user interfaces, and web data access.

Overview

This course is an introduction to Programming as a skill, a discipline, and a profession for graduate students. We'll dive into hands-on programming from day one and progress to evaluating, using, and contributing to open source libraries and frameworks. We'll focus equally on reading and writing code. Students will leave the course with real skills, an ability to learn new programming technologies, and an understanding of how to incorporate open source code into their projects. It will serve as an appropriate foundation for students seeking a career in programming and indispensable background for any information professional needing to evaluate, communicate with, or work with programmers or code.

Objectives

At the end of this course, students should:

Have the skills required to solve problems by creating and modifying programs and systems, using modern programming tools.

Have the knowledge of basic programming concepts, their appropriate usage, and how and where to learn more.

Have an attitude of confidence when reading, writing, or discussing computer code

Students will be prepared to integrate these skills and knowledge with other SILS courses in Databases, Web Development, Text Mining and others.

Prerequisites

There are no prerequisite course requirements for the course. However, you will need to be skilled in the use of basic mathematics and algebra, as well as, email and web usage.

Textbook

Think Python. Available free on the web at
http://www.greenteapress.com/thinkpython/.

Software

We will be expecting you to use Pycharm Community Edition Integrated Development Environment (IDE) and Python 2.7.8. You can download and install Pycharm Community Edition IDE at no cost from http://www.jetbrains.com/pycharm/download/. We will discuss how to use the Pycharm Community Edition IDE in class.

Hardware

Students will need their own laptops. Bring your laptops to class everyday. Computers conforming to CCI guidelines should do fine. Contact me if you have any questions.

Attendance

Attendance is mandatory for all weekly lectures. If you have a scheduling conflict, you should take this course another semester. No attendance will be taken but the instructor will not repeat material covered in class.

Course Web Page

http://ils.unc.edu/courses/2014_fall/inls560_001/

Getting Help and Class Discussion

We will be using Piazza for class discussion and getting help. The system is highly catered to getting you help fast and efficiently from classmates and myself. Rather than emailing questions to me, I encourage you to post your questions on Piazza. If you do not get a response within a day or two on Piazza, please send mail to carterjl@cs.unc.edu . But try Piazza first.

Before posing a question, please check if this question has been asked before. This will reduce post clutter and reduce our burden. Repeat questions will be ignored by the

instructors.

Piazza allows anyone to respond. So if you see a question that you think you can respond to, please do so, as that will reduce our burden and help you "teach" your fellow students.

This will be a form of class participation that will be noted when I allocate my fudge points!

If you have any problems or feedback for the developers, email team@piazza.com.

Signup Link: piazza.com/unc/fall2014/560

Class page: piazza.com/unc/fall2014/560/home

We assume that you have a UNC electronic mail account and are familiar with using email and the World Wide Web. Email will be used extensively in this course. We will use to communicate to the class essential information concerning programming assignments. You will use it to ask questions and give feedback on the assignments and course. You should check your email at least five times per week. Every time you sit down at a computer to work on an INLS 560 assignment you should first check your email – there may be important information about the current assignment. The web page for this course will contain copies of many of the handouts, programming assignments, information on the reading assignments, etc.

We will be using Sakai alongside the course webpage to post PowerPoint lecture slides and other information.

Grading

Assignments	60%
Midterm	15%
Final Examination	25%

The above percentages add up to 100%, but I reserve the right to apply a 10% fudge factor to give consideration to things such as good class participation, stellar programs, extra credit, answers to exam questions that go far beyond what is expected, and bona fide extenuating circumstances.

There will be several programming assignments, ranging in duration from 1-3 weeks. The number of points per assignment will generally increase during the course of the semester reflecting the increased length and complexity of the assignments. Several of the assignments will build on previous ones. Each assignment will be due on a specific date; late assignments will not be accepted.

The criteria for the grading of the programming assignments are: Correctness and Style. Programs will be expected to be 100% correct. Programs must be readable by humans as well as machines. This requires that they be based on clear thought and be well presented.

Note that the SILS grading policy is based on the University Grading Policy. SILS uses the following graduate and undergraduate grading scales:

Graduate		Undergraduate
H	Clear excellence Entirely satisfactory Low Passing Failed	A
P	Entirely satisfactory	A-, B+, B, B-
L	Low Passing	C+, C, C-
F	Failed	D+, D, F

Making up Homework and Examinations

Serious illness, a death in the family, and activities such as intramural meets and student exchange programs can justify waiving or relaxing the usual rules for class work and examinations. (But the problems of student life, including the consequences of procrastination and commitments to other courses, cannot.) In circumstances that merit special treatment, documentation is usually available to the student, and I feel most comfortable when a request for special consideration is accompanied by appropriate written material supporting the request. In cases where events that will interfere with course work are foreseen, a student should discuss the matter with me well before the work is due.

Incompletes

Incompletes will be given only in dire emergencies. Documentation (such as a physician's note) will generally be required. Falling behind in your work is not an emergency.

Courtesy

We will try to be courteous to you; we ask that you be courteous to us. Please do not read the newspaper or other materials during main lecture or section meeting; or work on your laptop on anything but looking at the class notes or taking your own notes of the lecture. If you must be late once or twice, take an aisle seat quietly; likewise if you must leave early. If this becomes habitual, you should drop the course. Most importantly, please remain quiet, except, of course, to ask questions; private discussions between students, even whispers, carry surprisingly well and are a real distraction to those seated near you and to the instructors. Thank you.

Cooperation and Honor Code:

You are encouraged and expected to discuss the material in class and all assignments among yourselves.

You are permitted to discuss all aspects of the Python programming language with anyone.

You are encouraged and permitted to discuss and cooperate on all written assignments, but you are expected to understand all material that you submit. Examinations will consist primarily of material similar to that of the assignments.

You are encouraged to discuss all programming assignments, but not your solution to the assignments.

How to Succeed in This Course

Participate in Class

Class participation has been a big factor in my upgrading borderline cases.

Review after Class

Review Powerpoint material and readings after each class if they have been distributed. A good self-test is: Can I explain the material to someone else?