

Syllabus

INLS 560: Programming for Information Professionals

Instructor: Prof. Elliott Hauser

Official Course Description

INLS 560: Programming for Information Professionals (3 credits)

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. Offered fall and spring.

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. Students will also be well versed in common tools and workflows used by developers and development teams.

Real-world Tools

This class will introduce you to and utilize many of the tools that real development teams use for collaboration,

coding, and communication. This will include Github, Jekyll, terminal and GUI-based text editors, videoconferencing, persistent online chat, and screensharing. By the end of the course you should be comfortable joining and collaborating on real development teams.

Flipped Design

I'd like to emphasize the flipped design of the course. Almost all of the content consumption (readings, lectures, etc) will be pushed outside of the class sessions. During class we'll focus on completing exercises almost every day, which will be due by midnight if you need extra time (see below). I'll help answer questions and provide clarification, but understanding the concepts and techniques in the content I assign you will be essential for completing the exercises in class. Failure to do this will be a serious hindrance to understanding programming, completing the exercises and doing well in the class.

Assignments

Cultural Immersion Activities: Like other cultural products, digital technologies require an immersive experience for effective learning. After a vocabulary of about 300-400 words is developed, students can begin teaching themselves languages, and a roughly similar threshold applies to programming concepts and technologies. Students in INLS 560 will immerse themselves in the culture surrounding programming in three ways:

- Weekly reading and sharing of technical blogs, news outlets, and question-answer sites
- Attending three in-person programming meetups over the course of the semester
- Discussions with our guest speakers from industry

These experiences will combine with class assignments, readings, and in-class exercises to deepen students' interest and facility with the concepts and skills we'll learn. Students who are unable to attend the in-person meetups may propose a comparable virtual community interaction in its place.

In-Class Exercises: Almost every class will feature exercises that build on assigned readings. These assignments can be completed and submitted during class, or students can submit them before midnight if they need extra time. Exercises may involve pair programming and I will circle through class to resolve any confusion and share important concepts with the group. Some of these exercises will involve code submission via Github, while others will be online exercises. In almost all cases you'll report your work or turn it in via Github.

Readings & Jekyll Blog: Assigned readings, lectures, or exercises should be completed before the indicated class. In most cases their content will be essential to completing the in-class exercises. In technology we'll often encounter terms or project names that we haven't heard of before (myself included). We'll collaboratively author a Jekyll blog on Github, at first as an in-class exercise, and then as a way to keep explain these novel concepts and projects to each other and the community at large.

Google Plus Community: We'll use a private Google Plus community for helping each other with assignments and sharing interesting readings. We'll be able to utilize the Hangouts feature for informal discussion via video and for pair programming via screensharing.

Final Project: The semester will culminate in a fully developed program or system of the student's creation. The project should utilize one or more prominent open source library or framework and be fully documented. Projects

can be selected from examples to be provided by the instructor or the student can design their own with approval from the instructor. Due to the interconnected nature of modern programming, projects that utilize code libraries from multiple students will be given extra credit and be eligible for an H grade (see below). The Final Project is due at 11:59 pm on Saturday, May 3rd. See below for my policy on late work (short version: don't be late).

Extra Credit: There will be infrequent opportunities to earn small amounts of extra credit. This will usually involve doing things I might normally do, like deploying our class blog, tidying up code I write for this class for wider consumption, or helping moderate our forums.

Materials

Readings: Most, if not all, materials used in this course will be drawn from freely available resources on the Internet. This mimics the working environment of most programmers. I'll select readings from blogs, project documentation, and leading free textbooks and developer reference books Think Python, Learn Python the Hard Way, and Pro Git.

Software: The course will utilize free open source software almost exclusively. For in-class exercises, some exercises will be completed via a browser and for actual development we'll all use the same version of Ubuntu Linux to eliminate configuration and compatibility issues. Students will be able to choose between installing Ubuntu on a VirtualBox virtual machine and installing the operating system on their own hardware. VirtualBox is a free, open source project that allows Mac, Windows, and Linux systems to host other operating systems. Detailed walkthroughs will be provided for all software installations and there will be time in class for troubleshooting.

Hardware: Students will need their own laptops. Computers conforming to CCI guidelines should do fine. Older hardware or netbooks might have trouble with virtualization, so please contact me if you'd prefer to set up a dual-boot system. My 2010 MacBook Pro runs VirtualBox just fine, and a native installation on an old 2007 MacBook works great as well. Contact me if you have any questions.

Grading

Weights of course components are as follows:

- Cultural Immersion Activities: 10%
- In-Class and Google Plus Community Participation: 20%
- In-Class Exercises: 30%
- Final Project: 40%

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	A
P	Entirely satisfactory	A-, B+, B, B-
L	Low Passing	C+, C, C-
F	Failed	D+, D, F

For the purposes of this course, satisfactory performance (the 'P' grade) involves completing all of the assignments and showing personal mastery of the material. Clear excellence (the 'H' grade) means participation in communities and effective collaboration with other programmers, either inside or outside our class.

Student and Instructor Expectations

Honor Code

Students should familiarize themselves with the University of North Carolina at Chapel Hill Honor Code which is described in University publications. For this course in particular, collaboration with other students will be expected in many instances, but work that is assigned to you as an individual should be yours alone. In group projects a combination of git commit histories, writeups and/or project documentation should attribute individual and collaborative work.

Attendance & Attention

Students should attend every class and give it their undivided attention. Use of non-coding websites or social features is prohibited. There will be many opportunities for interaction, but during class time please pay attention.

Late Work

In general, lateness is not tolerated in the professional world, and these are the standards I'm holding you to. Late work without prior approval will not be accepted. Contact me as soon as you think you may have difficulty meeting a deadline with your rationale for an extension. Turning work in early is good for all involved, for this class and your career beyond.

Diversity & Inclusiveness

I take the happiness and wellbeing of all students very seriously and share the university's commitment to diversity (available here: <http://diversity.unc.edu/our-committment/div-values/>). The field of programming has historically struggled with diversity but we have the opportunity to help build a more inclusive future for the profession. I expect students to join me in welcoming different perspectives and backgrounds in the course.

Acknowledgements

This course and its content has been deeply influenced by my interactions with and the input of others. I'd like to thank Hilmar Lapp, Stephanie Haas, Brian Marks, Greg Wilson, John D. Martin III, and Alan Dipert for a wide range of help.

INLS 560 - "Programming for Information Professionals"

Another [silshack](#) by [@hauspoor](#) & friends

This page was generated on Thursday, April 09, 1914 2:48:08 PM