LIN 4930/6932 Thinking in Code: Linguistic Data

Spring 2026

MWF 12:50-1:40 Room: TUR 2353

Sarah Moeller
Email: smoeller@ufl.org
Tel: (352) 294-7449
Office: Turlington 4129
Office hours: M 9:30-11:30am

Borui Zhang
Email: boruizhang@ufl.edu
Tel: (352) 273-8434
Office: Communicore C2-203A
Office hours: F 2:00-3:00pm

Course description

Computational thinking and programming skills are important practical tools in linguistics. Students will learn how data structure choices and design shape outcomes in computational approaches to linguistics, using computer code to preprocess raw data and represent linguistic analysis through a rule-based finite-state transducer. LLMs will be used as supportive and learning tools, to write, explain, and troubleshoot code, accelerating the learning process without requiring model training. The course emphasizes problem solving and aligning technical implementations with linguistic theory. The material assumes students have skills and knowledge covered in LIN 4932/6930 Intro. Python Programming and LIN 2010 Intro. to Linguistics.

Goals

After the course, students will able to:

- . . . use optimal Python data structures for processing linguistics data.
- . . . code computationally efficient algorithms.
- . . . use large language models (LLM) as coding assistants.
- . . . build AI parsing and word generation tools.

Grade breakdown

Participation and Attendance	10%
Homework + Final Project (LIN 6932)	50%
Quizzes	25%
Presentations	15%

Participation and Attendance

Students will do coding exercises in class. These may be graded for participation.

Homework

Take home assignments must be submitted by the due date. Graduate students submit a final project.

Quizzes

Quizzes cover concepts and terminology and coding knowledge. Readings will be available on Canvas and may be quizzed in class.

Presentations

Students will present their work in class. They will be graded on professional oral presentation skills. Presentations should include slides and/or handouts to aid the audience's understanding.

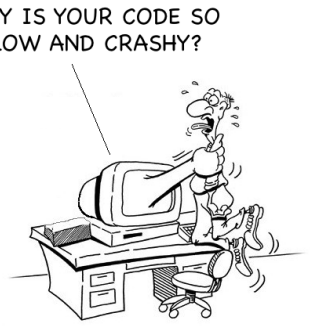

Figure 1: paperperweek.wordpress.com

The Fine Print

All class policies adhere to the UF syllabus policies: https://syllabus.ufl.edu/syllabus-policy/uf-syllabus-policy-links/

CLASSROOM CONDUCT. Students and faculty each have responsibility for maintaining an appropriate learning environment. Those who fail to adhere to such behavioral standards may be subject to discipline. I pledge to treat each of you with dignity, respect, and professional courtesy; I expect you to do the same for me and for each other.

RELIGIOUS OBSERVANCES A student should inform the instructor of observances of their faith that will conflict with class attendance, tests or examinations, or other class activities *prior* to the class or occurrence of that test or activity. Faculty is obligated to accommodate that particular student's religious observances. See policy details at https://catalog.ufl.edu/UGRD/academic-regulations/attendance-policies/#religiousholidaystext.

COLLEGIAL ENVIRONMENT. Students are encouraged to employ critical thinking and to rely on data and verifiable sources to interrogate all assigned readings and subject matter in this course as a way of determining whether they agree with their classmates and/or their instructor. No lesson is intended to espouse, promote, advance, inculcate, or compel a particular feeling, perception, viewpoint or belief. Students are encouraged to share their viewpoints, data, and sources in class and to speak with the instructor or classmates, in class or privately, about any perceived violation of this policy.

GRADING SCHEME. This course follows UF grades and grading policy:
https://catalog.ufl.edu/UGRD/academic-regulations/grades-grading-policies/

		B+	87-89%	C+	77-79%	D+	67-69%		
A	94-100%	B	83-86%	C	73-76%	D	63-66%	F	0-59%
A-	90-93%	B-	80-82%	C-	70-72%	D-	60-62%		

SCHEDULE (subject to change)

WEEK	TOPIC	DUE
1	Coding Environment; Data Structures; `xml` library	
2	Computational Complexity	
3	Algorithm Analysis	
4	Preprocessing Linguistic Field Data	HW 1
5	Introduction to LLMs and Generative AI	
6	Accessing LLMs	
7	Vibe Coding and Processing Datasets	
8	Semantic Parsing	
9	Syntactic Parsing	HW 2
	SPRING BREAK	
10	Finite State Automata and transducers (FST)	
11	FST morphology	
12	`pyfoma`	
13	`pyfoma` cont'd	HW 3
14	Student Presentations	
	FINALS	Final Projects