Ling 4400/4020: Computational Language Processing

Course Number: LING-4400 (graduate), LING-4020 (undergraduate)
Lecture: Monday and Wednesday, 12:30 - 1:45
Location: ICC 207b

Instructor: Ethan Gotlieb Wilcox
Office: Poulton 248 (Office Hours: Wednesday, 2:00 – 3:00 pm)
Email: ethan.wilcox@georgetown.edu
Website: https://wilcoxeg.github.io/

Teaching Assistant: Xiulin Yang
Office Hours: TBD
Email: xy236@georgetown.edu

Summary: This course will introduce students to the basics of Natural Language Processing (NLP), a field that combines linguistics and computer science to produce applications, such as generative AI, that are profoundly impacting our society. We will cover a range of topics that form the basis of these exciting technological advances and will provide students with a platform for future study and research in this area. We will learn to implement symbolic representations such as finite-state techniques, *n*-gram models, and basic parsing in the Python programming language. Previous knowledge of Python is not required, but students should be prepared to invest the necessary time and effort to become proficient over the course of the semester. Students who take this course will gain a thorough understanding of the fundamental methods used in natural language processing, along with an ability to assess the strengths and weaknesses of natural language technologies based on these methods.

This course fulfills the college's *Quantitative Reasoning and Data Literacy* (QRDL) requirement as part of the Core Curriculum.

Learning Outcomes:

- Students will learn how to write code in Python
- Students will learn how to debug Python code using a debugger
- Students will become familiar with the theoretical fundamentals of programming
- Students will become familiar with the fundamental techniques used in natural language processing, including string manipulation, finite state techniques, language modeling, and vector space models
- Students will gain practice thinking critically about the behavior, abilities, and limitations of current NLP tools, including large language models

Course Schedule

Week	**Module**	**Assignments Out & Due**	**Readings** (all sections are inclusive)

Jan 5, 2026 (1 session, Jan 7th)	Introduction	1/7 Assignment 1 (installing Python)	
Jan 12, 2026	Intro to Python, String Manipulation	1/14 Assignment 1 due 1/14 Assignment 2 (palindrome checker)	
Jan 19, 2026 (1 session, Jan 21st)	Tokenization, Regular Expressions		Jurafsky & Martin, Chapter 2.0 - 2.2; 2.5 - 2.7
Jan 26, 2026	Regular Expressions, Finite State Automata	1/26 Assignment 2 due 1/26 Assignment 3 (phone scraper)	Jurafsky & Martin (2007 edition) 2.2 - 2.2.4 (PDF available in Canvas)
Feb 2, 2026	Finite State Automata, Finite State Transducers	2/4 Assignment 3 due 2/4 Assignment 4 (regular expressions)	Jurafsky & Martin (2007 edition) 3.0 - 3.5 (PDF available in Canvas)
Feb 9, 2026	Introduction to Probability, Language Modeling		Jurafsky & Martin, Chapter 3.0 - 3.1
Feb 16, 2026 (Monday session held on Tuesday, Feb 17th)	Advanced Language Modeling	2/18 Assignment 4 due	Jurafsky & Martin, Chapter 3.2 - 3.6 (3.7 is a fun bonus if you're interested!)
Feb 23, 2026	Midterm review **Wednesday, February 25th: Midterm Exam (in class)**		
Mar 2, 2026	Spring Break! (No classes)		
Mar 9, 2026	Intro to Neural Network Language Models, Introduction to Markov Models	3/9 Assignment 5 (language modeling)	Jurafsky & Martin, Chapter 6.0 - 6.5
Mar 16, 2026	Sequence Tagging	3/18 Assignment 5 due 3/18 Assignment 6 (part of speech tagging)	Jurafsky & Martin, Chapter 17.0 - 17.3
Mar 23, 2026	Named Entity Recognition, Introduction to Context-Free Grammars	3/25 Assignment 6 due 3/25 Assignment 7 (named entity recognition)	Jurafsky & Martin, Chapter 17.4
Mar 30, 2026	Context-Free Grammars and Parsing		Jurafsky & Martin, Chapter 18.0 - 18.5 (You can skip 18.4)
Apr 6, 2026 (1 session, April 8)	Vector Space Modeling	4/6 Assignment 7 due 4/6 Assignment 8 (vector space models)	Jurafsky & Martin, Chapter 5.0 - 5.4

Apr 13, 2026	Vector Space Models	4/15 Assignment 8 due	Jurafsky & Martin, Chapter 5.4 - 5.9
Apr 20, 2026	Naive Bayes Classifiers	4/22 Assignment 9 (practice final; not graded for correctness)	Jurafsky & Martin, Appendix B.0 - B.8
Apr 27, 2026 (1 session; April 27th)	Final Review	4/29 Assignment 9 due	
May 4, 2026	**Final Exam: (12:30 - 2:30 pm)**		

Prerequisites: There are no prerequisites for this course. In particular, we will assume that students have no prior experience with programming. If you *do* have some limited prior experience with coding but not with NLP, this course is still likely appropriate for you. However, please speak with the course instructors if you have any questions or concerns.

Grade Breakdown

Midterm Exam	25%
Final Exam	35%
Homework Assignments	30%
Participation and Attendance	10%

Graduate vs. Undergraduate Section: Requirements for LING 4400 and LING 4020 are largely the same, with the following differences:

- Students enrolled in 4400 will be asked to complete additional questions on the homework assignments. Students in 4020 can complete these for extra credit, with a max homework grade of 100% (that is, additional points earned on homework won't "spill over" to boost midterm or final grades).
- Midterm and final exams will differ slightly between the sections, with slightly more challenging material for the 4400 exam. The number of questions and total number of points will remain the same between the exams.

Participation and Attendance: This course is a challenging, fast-paced introduction to two different topics, computer programming and natural language processing. Given that we will be moving quickly between modules, it is very easy to fall behind. I encourage you to miss class only for things such as planned medical events, medical emergencies, family emergencies, or religious observations. While we will not be keeping track of attendance formally, 10% of your grade will be assigned based on your participation. Participation can take many forms – asking questions in class, engaging in group activities, and attending office hours. For more information regarding the university policy on attendance, please see the academic standards section of the Undergraduate Student Bulletin (https://bulletin.georgetown.edu/regulations/standards/)

- Class time will involve personal and group coding sessions, so **please bring your laptop to class!**

Assignments: Over the semester, students will complete eight programming assignments and one practice final (graded for completion only). Assignments are due before the start of class on their specified due date. Submissions will be graded for correctness by the TA. One week after the assignment is due, we will review the solution for the assignment in class. Due dates for the original submission are given in the above course schedule; however, they may be adjusted throughout the semester. All submissions happen through Canvas.

- **Late Policy:** Late homework can receive credit, but the grade you can achieve is capped based on how late it is, with the maximum grade you can achieve lowered by 10% per 24-hour period after the due time. Once we review the assignment in class, submissions are no longer permitted.

- **Group Work Policy:** The code that you submit as part of homework assignments *must be written by you*. However, working in groups on assignments can be extremely helpful. If you work with a group to come up with *pseudocode* (a high-level plan for how your program will run), or if you work with a group to understand and debug a piece of non-functioning code, that's OK. But you need to implement the pseudocode, or fix the bug yourself!

For more information on the University's honor code, please see https://honorcouncil.georgetown.edu/.

Readings: Most of the course readings will be derived from Jurafsky and Martin, 3rd edition, which can be accessed online here: https://web.stanford.edu/jurafsky/slp3/. There are several readings from an older version of this text, which will be posted online in Canvas. Readings *not* from this book will also be posted online.

Midterm and Final: The midterm and final exams will not require live coding on a computer. Rather, they will involve commenting on or correcting code, explaining why and how code does (or doesn't) work, as well as explaining concepts and working with formulae we've covered in class. The TA will lead practice sessions in advance of each exam. The mid-term will take place during a class session.

Use of AI Assistants: The use of AI assistants, such as ChatGPT, Bard, or Claude, as well as coding plug-ins like Copilot, are not explicitly forbidden. These tools are increasingly part of our world and are based on the technologies we will learn about in this class. Indeed, one of the learning outcomes is to encourage critical thinking about the abilities and limitations of such systems.

With that in mind, a warning: While LLM-based tools can be extremely effective, they suffer from two major drawbacks. First, they make mistakes. Second, they reduce the amount *you* have to think about and understand the code you run (that's the point!). If you are an advanced coder or an expert debugger, then you will be able to spot the mistakes that these systems make and fix them. However, for beginners, it's quite possible that correcting a buggy LLM-generated algorithm will take more time than writing the algorithm yourself. Furthermore, every time you rely on an LLM to generate something for you, that's one less opportunity you have to solidify the concepts we learn in lectures and to practice coding as a skill. Importantly, you won't be able to use LLMs during the midterm or the final. If you use these systems extensively in the beginning, you'll have a much harder time preparing for the exams.

Accommodations: If you have a recognized accommodation through ARC, please contact Ethan. More information about accommodations and support, including student-athlete support, can be found on the ARC website (https://academicsupport.georgetown.edu/)