```
#!/bin/bash
# Downloads Universal Dependencies data for the Europarl languag

set -euo pipefail

organize() {
  echo -n "Organizing ${1}..."
  local -r DATADIR="data/${2}"
  mkdir -p "${DATADIR}"
  mv ud-treebanks-v2.1/UD_${1}/${2}-ud-dev.conllu "${DATADIR}"
  mv ud-treebanks-v2.1/UD_${1}/${2}-ud-test.conllu "${DATADIR}"
  mv ud-treebanks-v2.1/UD_${1}/${2}-ud-train.conllu "${DATADIR}
  echo "done"
}

echo -n "Downloading tarball..."
curl -so ud-treebanks-v2.1.tgz "https://lindat.mff.cuni.cz/repo
11234/1-2515/ud-treebanks-v2.1.tgz?sequence=4&isAllowed=y"
echo "done"

echo -n "Decompressing tarball..."
tar -xzf ud-treebanks-v2.1.tgz
rm ud-treebanks-v2.1.tgz
echo "done"

mkdir -p data
organize "Bulgarian" "bg"
organize "Czech-CAC" "cs_cac"
organize "Czech-CLTT" "cs_cltt"
```

LING78100/73800: METHODS IN COMPUTATIONAL LINGUISTICS I

FALL 2022

CUNY GRADUATE CENTER

Instructor: Prof. Kyle Gorman
Practicum leader: Cameron Gibson
Lecture: Tuesday 4:15-6:15, location TBD
Practicum: Friday 4:15-6:15, location TBD
Office hours: Monday 2-4, GC 7400.01 and by request

SYNOPSIS

This course is the first of a two-semester series introducing modern software development. The intended audience are students interested in speech and language processing technologies, though the materials will be beneficial to all language researchers.

OBJECTIVES

Using the Python programming language, students will be able to write programs which count the frequencies of various linguistic phenomena in text. They will be able to process text stored in various *structured data* formats. They will come to understand how computers encode multilingual text. They will

MATERIALS

Some readings will be assigned. Students are strongly encouraged to bring a laptop computer to the lecture and practicum. Students are also welcome to use the Computational Linguistics Laboratory (7400.13) for practice and assignments.

ASSIGNMENTS

Assignments will take the form of a small software development projects accompanied by a write-up describing the general approach taken and any challenges encountered. Students will usually be able to verify the technical correctness of their code by running a provided unit test. Students will also be graded on the readability of their code, and the quality of the write-up. We will use GitHub Classroom for assignment turn-in.

The final assignment will be an open-ended project which will involve collecting basic statistics (e.g., counts) of some linguistic phenomenon from either raw text or structured data. Students are encouraged to conceive of projects relevant to their research interests. Students should discuss project plans with the instructor during office hours to confirm that it is both *feasible* and of *appropriate scope*. Because of the open-ended nature of the final assignment, unit tests will not be provided.

GRADING

80% of students' grades will be derived from the assignments; the remaining 20% will be reserved for participation and attendance. Assignments must be submitted on time or will receive a 0 grade (barring a documented emergency).

ACCOMMODATIONS

The instructor will attempt to provide all reasonable accommodations to students upon request. If you believe you are covered under the Americans With Disabilities Act, please direct accommodations requests to Vice President for Student Affairs Matthew G. Schoengood.

ATTENDANCE

Students are extended to attend all lectures and practica. The instructor reserves the right to tie grades to attendance records. The instructor and practicum leader are not responsible for reviewing materials missed to absence.

INTEGRITY

In line with the Student Handbook policies on plagiarism, students are expected to complete their own work. However, a student is permitted to collaborate with another student during the coding phase of an assignment so long as they: do not share lines of code with each other, *mutually* disclose their collaboration in their write-ups, and do not collaborate *at all* on their write-ups.

The instructor reserves the right to refer violations to the Academic Integrity Officer.

RESPECT

For the sake of the privacy, students are asked not to record lectures. Students are expected to be considerate of your peers and to treat them with respect during class discussions.

SCHEDULE

(Please note that this is subject to change.)

8/30		Syllabus and motivations	Bird et al., ch. 1
9/6		Literals; variables; operators	
9/13	HW1 due	Control flow	
9/20	HW2 due	Indexing; slicing; sorting	Sorting HOW TO
9/27	No class		
10/4	No class		
10/11	HW3 due	Functions	
10/18	HW4 due	Containers	Kuchling, `collections`
10/25		File I/O	
11/1	HW5 due	Generators	
11/8		Classes	
11/15	HW6 due	Text encoding	Bird et al., ch. 3.3, Gorman, Spolsky, `chardet`, `unicodedata`
11/22		Modules; random numbers	Bird et al., ch. 3-3.2, `random`
11/29	HW7 due	Command-line design; CSVs	`argparse`, `csv`
12/6		Regular expressions	Bird et al., ch. 3.4, `re`
12/13	HW8	NLTK	Bird et al., ch. 5, Church

due

REFERENCES

Bird, S., Klein, E. and Loper, E. n.d. *Natural Language Processing with Python*. URL: https://www.nltk.org/book/.

Church, K. W. No date. Unix™ for poets. URL: http://doc.cat-v.org/unix/for-poets/kwc-unix-for-poets.pdf.

Jurafsky, D., and Martin, J. H. 2009. *Speech and Language Processing*. 2nd edition. Pearson. (See also: 3rd edition draft.)

Kuchling, A. 2007. Python's dictionary implementation: being all things to all people. In A. Oram and G. Wilson (ed.), *Beautiful Code*, pages 293-301. O'Reilly.