

Spring Semester 2018
Iowa State University

ENGL 516 - Formal Methods of Linguistic Analysis

Course Handbook

Instructor: Sowmya Vajjala

- *Office:* 331 Ross Hall
- *Email:* sowmya@iastate.edu

Course Objectives: Learning how to program is a useful skill in today's world, given the penetration of information technology into diverse disciplines. This is especially true in Applied Linguistics, as the use of language processing tools in and outside classrooms is increasing day by day. Apart from adapting to the technology, learning to do tweak software tools you use at work will enable you to customize existing technologies to your needs. This will also let you enhance the technologies with new features that are derived from your domain expertise. This course serves the purpose of empowering applied linguists in this direction.

In this course, we will discuss basic concepts required to write your own computer programs using a high-level programming language (This iteration of the courses uses Python as that language). We will discuss issues such as - how to read data (e.g., text) into a computer program, process the data and get what we want out of it, store and display the results etc. In a short detour, we will also touch upon text processing tools that are inbuilt in Unix machines (MacOS, Linux etc).

Learning Outcomes: Upon successful completion of this course, students are expected to be able to do the following:

- Understanding how text is represented on computers, and how to work with it using regular expressions
- Given a problem description, think about an approach to solve the problem, design and write computer program(s) to implement the approach
- Know where to look for to not reinvent the wheel. Read the documentation for existing code and write their own programs extending them.
- Understand how to find and fix errors in your programs (debugging)

Some concrete expectations for this particular iteration of the course are that the students will be able to:

- Write small programs that involve reading, processing and writing content into files on the computer

- Implement simple browser based applications involving collecting and displaying data to the user

Pre-requisites: Familiarity with using computers and an interest in learning how they work and how to make them do what you want.

Course Details:

- meets in Ross 312, on Tuesdays and Thursdays from 12:40-2 pm
- *Office hours:* Tuesdays and Thursdays, 2-3 pm (please email beforehand if there are specific issues to discuss)

Credits:

- Credit Points: 3
=> (3 hours of classroom instruction or 5-6 hours of lab instruction) + (6 hours of additional effort on student's end). Please note: It will take longer than you think to complete tasks in this course. It is different from other kind of courses you are used to.

Nature of the course and expectations: This is a 3 credit course and is meant to be an introductory course. Primary mode of instruction are lectures, discussion and hands on programming tasks. We will have regular assignments, a final project and an oral exam for the project. Readings for each topic are specified in the syllabus and it is expected that the students read them BEFORE coming to the class. There may be a few additional (mostly optional) readings from other sources for some of the topics.

Students enrolled in the course are expected to:

1. regularly and actively participate in class and answer questions posed in the discussion forum (5% of the grade)
2. submit the assignments on time (65% of the grade)
3. finish a programming project as a final exam for the course and attend an oral examination about it (30% of the grade)

Grading Policy There are 6 assignments with 10% weightage each and 1 assignment for 5%. They come once a week to three weeks and usually have 2 weeks of time for submission. For the final exam, you have to implement a small group programming project, which is for 30% of your grade. The remaining 5% is for active classroom participation. You will be put into groups of 3 students (you can give your choices) before spring break, and all subsequent submissions are group submissions i.e., last three assignments and the final project will be group submissions. The project will be decided around mid-semester from a list of given topics. You can also choose your own topic after consulting me. The final exam grade will also have a oral defense session about your project which can happen in the class or in my office. Plus/minus grading will be used (93% = A, 90% = A-, 87% = B+, 83% = B, 80% = B-, etc.).

Deadlines

1. Assignment 1 (5%): January 20, 2018
2. Assignment 2 (10%): February 3, 2018
3. Assignment 3 (10%): February 17, 2018
4. Assignment 4 (10%): March 3, 2018
5. Final Projects - team formation and project selection: March 10, 2018
6. Assignment 5 (10%): March 24, 2018 (Group)
7. Assignment 6 (10%): April 7, 2018 (Group)
8. Assignment 7 (10%): April 14, 2018 (Group)
9. Final Project - presentation (10%): April 26 2018
10. Final Project - submission (20%): May 1st 2018
11. Classroom participation and posting on forum (5%): Every week

Attendance Policy: I don't take attendance, but you are recommended to attend the classes to be able to do the assignments.

Class etiquette: Please do not read or work on materials for other classes in this class. Come to class on time and do not pack up early. Electronic devices like mobile phones, tablets etc should not be used in the class. Laptops should not be open in class unless there is a concrete, assigned activity. If for some reason, you must leave early or you have an important call coming in, or you have to miss class for an important reason, please let me know (via email) and get it approved *before* the class. Being absent from the class does not allow you to skip submitting any assignments that were assigned in that class. Please pay attention, make notes.

Academic Conduct: Generally, you are encouraged to work in groups, discuss, and exchange ideas. At the same time, you are expected to do your assignments by yourself and with honesty. For the text you write, you always have to provide explicit references for any ideas or passages you reuse from somewhere else. Note that this includes text taken from the web. You should cite the url of the web site in case no more official publication is available. Specifically, the class will follow Iowa State University's policy on academic dishonesty. Anyone suspected of academic dishonesty will be reported to the Dean of Students Office: <http://www.dso.iastate.edu/ja/academic/misconduct.html>

Disability Accommodation Iowa State University complies with the Americans with Disabilities Act and Sect 504 of the Rehabilitation Act. If you have a disability and anticipate needing accommodations in this course, please contact (instructor name) to set up a meeting within the first two weeks of the semester or as soon as you become aware of your need. Before meeting with (instructor name), you will need to obtain a SAAR form with recommendations for accommodations from the Student Disability Resources, located in Room 1076 on the main floor of the Student Services Building. Their telephone

number is 515-294-7220 or email disabilityresources@iastate.edu . Retroactive requests for accommodations will not be honored.

Harassment and Discrimination Iowa State University strives to maintain our campus as a place of work and study for faculty, staff, and students that is free of all forms of prohibited discrimination and harassment based upon race, ethnicity, sex (including sexual assault), pregnancy, color, religion, national origin, physical or mental disability, age, marital status, sexual orientation, gender identity, genetic information, or status as a U.S. veteran. Any student who has concerns about such behavior should contact his/her instructor, Student Assistance at 515-294-1020 or email dso-sas@iastate.edu, or the Office of Equal Opportunity and Compliance at 515-294-7612.

Dead Week Policy This class follows the Iowa State University Dead Week policy as noted in section 10.6.4 of the Faculty Handbook: <http://www.provost.iastate.edu/resources/faculty-handbook>

Syllabus - topics covered Primary Textbook: Other useful readings and websites: listed in resources.pdf which can be seen in the same folder where you found this document.

1. Introduction

- Introduction to computing. How does a computer work? What is programming? Why python?
- Programming basics: Algorithm, flowchart, program.

Readings: Chapter 1 from the text book.

2. Python fundamentals

- Installing python on your personal machines/lab machines. Writing a hello world program
See this link for instructions: <https://www.py4e.com/install>
- Installing PyCharm Edu IDE for a user-friendly programming interface: <https://www.jetbrains.com/pycharm-edu/>
- Writing basic variable declarations, performing arithmetic operations
Readings: Chapter 2 in the textbook.

3. Writing python functions, running python code

- Conditional statements
- Loop statements
- Repetitions and iterations
- Writing python code using these concepts.
- Understanding and fixing bugs and errors in code

Readings: Chapters 3, 4, 5 in the textbook.

4. Python Data structures

- Strings, Regular Expressions with Strings
Readings: Chapters 6 and 11
 - Lists and Tuples
Readings: Chapters 8 and 10
 - Dictionaries and Hashing
Readings: Chapter 9
5. Reading from and writing into text files
Readings: Chapter 7
 6. Working with HTML files.
Readings: Chapter 12
 7. Understanding program flow
 - good practices to write our code
 - reading other people's programs
 - using python packages in our program and APIs
 Readings: Chapters 12,13; <http://www.nltk.org/book/ch04.html>
 8. Reading from a database and writing back.
Readings: Chapter 14
 9. Misc: Automating common tasks on computer
 - Replicating unix command functions
 - Search/Sort operations on text files
 Readings: Chapter 16 and Unix for Poets free ebook.
 10. Misc: Managing different versions of code. Introduction to Git.

Scheduling and Deadlines (tentative) Note that the following session plan is subject to change; it only constitutes the current state of our planning.

1. Tuesday, January 9: Course orientation and Introduction
(Old students may come to chat about their experiences)
2. Thursday, January 11: Introduction to computing. What is a program? Algorithm, Flowcharts etc.
Assignment 1 assigned. (5%)
Optional readings:
 - https://en.wikipedia.org/wiki/Computer_program
 - Chapter 2 in "Introduction to Computing" by David Evans
<http://www.computingbook.org/Language.pdf>
 - Chapter 3.1, 3.2 in the above book. <http://www.computingbook.org/Programming.pdf>

(Old students may come to chat about their experiences)

3. Tuesday, January 16: Downloading and installing Python and starting with python.
Optional Reading and Homework: <http://docs.python-guide.org/en/latest/starting/installation/>
4. Thursday, January 18: Basics: declaring variables, performing arithmetic operations
Readings: Chapter 2 in the textbook.
Assignment 1 Due on 20th January)
5. Tuesday, January 23: Writing conditional statements, indentation, catching exceptions
Readings: Chapter 3 in the textbook.
(Assignment 2 assigned (10%))
6. Thursday, January 25: Writing our own python functions.
Readings: Chapter 4 in the textbook.
7. Tuesday, January 30: Writing a pseudocode, Loops. Iteration. Recursion.
Readings: Chapter 5.
8. Thursday, February 1: Review of concepts so far.
(Assignment 3 assigned (10%). Assignment 2 Due on 3rd February)
9. Tuesday, February 6: Python Data Structures: Strings
Readings: Chapter 6.
10. Thursday, February 8: Regular Expressions and Strings
Readings: Chapter 11
11. Tuesday, February 13: Regular Expressions continued
12. Thursday, February 15: Reading and Writing data from Files
Readings: Chapter 7
(Assignment 4 assigned (10%). Assignment 3 Due on 17th).
13. Tuesday, February 20: Python Lists
Readings: Chapter 8
14. Thursday, February 22: Tuples and dictionaries in Python. Readings: Chapter 9 and 10.
15. Tuesday, February 27: data structures, file manipulations continued (with some hands on exercises and code analysis)
16. Thursday, March 1: Creating small web-based applications with Python - introduction
(Assignment 5 assigned (10%). Assignment 4 Due on 3rd March. Topics for final projects released).
17. Tuesday, March 6: Web applications, continuation
(Project/Rest of assignment groups decisions due: 10 March)
18. Thursday, March 8: Project ideas discussion + finalizing groups for remaining assignments and project
19. Tuesday, March 13: **Spring Break - no classes**
20. Thursday, March 15: **Spring Break - no classes**
21. Tuesday, March 20: Python and Databases - overview (Readings: Chapter 14)

22. Thursday, March 22nd: Databases continued.
(Assignment 6 assigned (10%). Assignment 5 Due on 24th March.)
23. Tuesday, March 27: Reading and understanding code, using somebody's programs for our purpose.
Readings: Chapters 12,13
24. Thursday, March 29: Working with HTML pages and parsing text content.
Readings: Chapter 12.
25. Tuesday, April 3: Hands on exercises (databases, using other Python API etc)
26. Thursday, April 5: Automating common tasks on your computer -
Readings: Unix for Poets - free ebook.
(Assignment 7 assigned (10%). Assignment 6 Due on 7th April)
27. Tuesday, April 10: Practice exercises, Optional: Version control: Git (overview)
28. Thursday, April 12: Practice exercises, Optional: Version control: Git (overview)
(Assignment 7 due on 14th April)
29. Tuesday, April 17: Revision, Hands on exercises etc.; Sorting and Searching- a few popular algorithms
30. Thursday, April 19: Revision, Hands on exercises
31. Thursday, April 21: Discussion about projects.
32. Tuesday, April 24: Oral Presentations, Revision.
33. Thursday, April 26: Oral Presentations, Wrap-up.
Project presentations Due (10%)
34. May 1st 2018: Exams week. Final project submission due **Final project due. (20%)**