PSY 341K:
Python Coding for Psychological Sciences
(Unique# 42811)
Spring 2018

Class Meets: 2-3:30pm, Tuesdays & Thursdays, SEA 2.122

Instructor: Satoru Hayasaka
Office: SEA 2.214
Email: hayasaka@utexas.edu
Phone: 512-475-6177

Office hours:
10-11:30am, Tuesdays & Thursdays
and by appointment

Course Description

***University Catalog Course Description*:** Introduction to coding with Python programming language, covering basic elements of programming for simple data processing, analysis, and visualization. A particular emphasis on numerical data in psychology and other scientific fields. No prior experience in programming or coding is required.

What will I learn?

Upon completion of this course, students will be able to:

- Understand basic concepts of programming
- Write simple python programs
- Troubleshoot errors during coding

Learning Outcomes:

1. Vocabulary and understanding of fundamental programming concepts
2. Python coding skills for simple data handling, manipulation, and visualization
3. Problem solving skills necessary to address a problem through Python coding

How will I learn?

This course will be presented as a hybrid of lectures and programming exercises. During each class, the instructor will present programming concepts and associated example codes in Python. The instructor will also give a number of small programming exercises throughout the class. Students will work on those exercises, and share their codes with the class in some cases.

There will be two mini-hackathons, or group programming projects, during the semester.

Pre-requisites for the course:

For psychology majors, upper-division standing and Psychology 301 and 418 with a grade of at least C in each; for others, upper-division standing, Psychology 301 with a grade of at least C, and one of the following with a grade of at least C: African and African Diaspora Studies 302M, Civil Engineering 311S, Economics 329, Educational Psychology 371, Electrical Engineering 351K, Government 350K, Mathematics 316, 362K, Mechanical Engineering 335, Psychology 317,Sociology 317L, Social Work 318, Statistics 309, Statistics and Data Sciences 301, Statistics and Data Sciences 302 (or Statistics and Scientific Computation 302), 303 (or Statistics and Scientific Computation 303), 304 (or Statistics and Scientific Computation 304), 305 (or Statistics and Scientific Computation 305), 306 (or Statistics and Scientific Computation 306), 318 (or Statistics and Scientific Computation 318), 321 (or Statistics and Scientific Computation 321), 328M (or Statistics and Scientific Computation 328M).

Course Resources

Required Resources

Computer: A computer with Python is required to complete in-class exercises, mini-hackathons, and homework assignments. Students are encouraged to bring a laptop to the class with Python and the required libraries installed. Those who do not have access to a laptop may use computers in the classroom. Although the instructor is willing to assist, installing Python on a laptop is ultimately each student's responsibility.

Python: We use Python 3 (3.5 or later) in this class. We DO NOT use Python 2 (any version), as some syntax is different between Python 2 and 3. Installation instructions for Python are covered during the first class.

Python libraries: Installation instructions for these libraries will be covered during the first class. Note that these libraries are for Python 3. Even if you have these libraries for Python 2, you need to install these for Python 3.

NumPy SciPy Matplotlib Pandas

Git: Git enables you clone (or copy) codes, notes, and data associated with this class from GitHub. Git is already pre-installed on Mac. For Windows users, you can download Git for free from its website (https://git-scm.com/downloads).

Optional Resources

Textbooks: There is no required textbook for this class. However, the first half of the class is loosely organized based on the materials from:

Al Sweigart,
Automate the Boring Stuff with Python:
Practical Programming for Total Beginners
No Starch Press (2015)

Jupyter Notebook: Jupyter Notebook enables you to read course notes and run Python codes on your web browser. Installation instructions and a brief demonstration of Jupyter Notebook will be presented during the first class.

Course Materials

Course Notes: Course notes are in Jupyter Notebook format, and available from the instructor's GitHub repository at:

https://github.com/sathayas/JupyterPythonFall2018

Students may clone the repository or download individual documents to be opened in Jupyter notebook on their computer. Alternatively, students can view the notes on their web browser. When viewed in a web browser, codes cannot be executed. Links to individual notes will be provided on Canvas. *Please note that notes will be added and updated throughout the semester.*

Codes & Data: Some example codes and data used during the class, as well as sample solutions for in-class exercises and homework assignments, will be available on GitHub at

https://github.com/sathayas/PythonClassFall2018

Students may clone the repository or download individual code or data file to be saved on their computer. Codes can also be viewed on a web browser. Codes are organized into folders for different modules. Links to folders are provided on Canvas.

Grading

Homework Assignments (60%)

There will be five homework assignments throughout the semester. Homework assignments will be posted on Canvas. In each assignment, students are asked to write Python codes to solve problems. Students are expected to turn in the source code(s) associated with each assignment on Canvas.

The four highest scores, out of five homework assignments (15% each), will be used to determine the final score.

Mini-Hackathons (30%)

There will be two mini-hackathons during the semester, each accounting for 15% of the grade. Each mini-hackathon is a group programming project by a team of 2-3 students. Two class periods will be devoted for each mini-hackathon. Students will be given a list of problems to be addressed prior to a mini-hackathon, and each student will choose a topic of interest. Those selecting the same topic will be organized into teams. The size of each team will be 2-3 students; if there is any uneven distribution of students, the instructor will merge / split teams into appropriate sizes. Each team will code collectively to address the problem, present their code and the solution to the problem, and submit the code.

Code Presentations (10+%)

Students present their code during in-class exercises. Students can post their code using the Discussion function on Canvas during a class, by responding to a post by the instructor. Among

posted submissions, one student is selected to explain his/her code to the class. Students can earn:

7% - First code presentation
3% - Second code presentation
1% - Third and subsequent code presentations

The instructor will ensure that different students have opportunities to present when selecting a submission. The priority will be given to students earning the most points for the code presentation. Depending on the number of presentations, a student may earn more than 10% for code presentations. The excess points will be added during the calculation of the final grade. If a student's percentage points for the final grade exceed 100%, then the grade will be capped at 100%.

Final Grade

The final grade is based on the percentage of possible points from the homework assignments, mini-hackathons, and code presentations (capped at 100%):

Percentage	Grade	Percentage	Grade
93-100	A	67-69.99	D+
90-92.99	A-	63-66.99	D
87-89.99	B+	60-62.99	D-
83-86.99	B	Below 60	F
80-82.99	B-		
77-79.99	C+		
73-76.99	C		
70-72.99	C-		

Policies

Attendance

Attendance is mandatory for the mid-term and final mini-hackathons. If you are unable to attend these due to religious holidays or extenuating circumstances (e.g., health issues, family emergencies), please consult the instructor as soon as possible so that an alternate arrangement can be made. The instructor will not track attendance for regular (non-hackathon) sessions. However, if you are not in the class, you will not be able to earn points for code presentations.

Late Assignments

Unless a student receives a prior permission from the instructor for a late submission, points will be deducted for assignments turned in late according to the following rule:

- Before the solution is presented in the class – 1%
- After the solution is presented in the class:

<1 day late	2%
2 days late	4%
3 days late	6%
More than 3 days late	8%
More than a week late	15%

If there is an extenuating circumstance (e.g., health problems, family emergencies, etc.) to

prevent you from submitting a homework assignment, *to avoid reduced points*, please ask the instructor's permission for a late submission ***BEFORE*** the deadline. When there is a late submission with a prior approval, the instructor may delay posting the solution on GitHub.

Group Submissions

Students may work together on a homework assignment. If students do work together, the students in the same group may submit identical codes. Each student is still responsible for submitting homework assignments individually on Canvas. When your homework assignment submission is part of a group submission, you must comment in your submission (either in the code itself or using the comment function in Canvas) whom you worked with. If nearly identical codes are submitted without noting any collaboration, such submissions are considered cheating. When students are working together, their group size may not exceed 4.

Academic Integrity

Plagiarism, cheating, collusion, and any other form of academic misconduct are considered a serious offence. The instructor will follow the definition of academic misconduct defined by Chapter 11, Student Discipline and Conduct, of the Institutional Rules on Student Services and Activities when determining whether such an infraction has taken place. Students can find materials on student conduct and academic integrity at

http://deanofstudents.utexas.edu/sjs/acint_student.php

Please note that the instructor will report any incident of academic misconduct. The sanction for an academic misconduct ranges from reduced points on assignments to dismissal from the university, depending on a student's prior records on academic misconduct.

Students with Disabilities and Different Learning Styles

This class respects and welcomes students of all backgrounds, identities, and abilities. If there are circumstances that make our learning environment and activities difficult, if you have medical information that you need to share with me, or if you need specific arrangements in case the building needs to be evacuated, please let me know. I am committed to creating an effective learning environment for all students, but I can only do so if you discuss your needs with me as early as possible. I promise to maintain the confidentiality of these discussions. If appropriate, also contact Services for Students with Disabilities, 512-471-6259 (voice) or 1-866-329- 3986 (video phone). http://ddce.utexas.edu/disability/about/

Course Schedule

***Note**: The course schedule is subject to change, depending on the amount of materials we can cover. It is your responsibility to note these changes when announced (although I will do my best to ensure that you receive the changes with as much advanced notice as possible).*

Date	Topic	Homework	Date	Topic	Homework
8/30	Introduction, Hello World		10/23	Mid-term mini-hackathon (Day 1)	
9/4	Operations, variables		10/25	Mid-term mini-hackathon (Day 2)	
9/6	Comparisons, if statement		10/30	Git and GitHub	
9/11	While loop		11/1	NumPy and arrays	HW 3 Due
9/13	For loop		11/6	NumPy and arrays (cont'd)	
9/18	For loop (cont'd)		11/8	Data visualization	
9/20	Functions		11/13	Data visualization (cont'd)	
9/25	Functions (cont'd)	HW 1 Due	11/15	Data frame with Pandas	HW 4 Due
9/27	Lists		11/20	Data frame with Pandas (cont'd)	
10/2	Lists (cont'd)		11/22	***Thanksgiving (no class)***	
10/4	Dictionaries		11/27	Data frame with Pandas (cont'd)	
10/9	String manipulation		11/29	GUI with Tkinter	HW 5 Due
10/11	Reading and writing files		12/4	Final mini-hackathon (Day 1)	
10/16	Reading and writing files (cont'd)	HW 2 Due	12/6	Final mini-hackathon (Day 2)	
10/18	Reading and writing files (cont'd)				