

ENGR-1330-PsuedoSyllabus

September 1, 2020

```
[1]: # Script block to identify host, user, and kernel
import sys
! hostname
! whoami
print(sys.executable)
print(sys.version)
print(sys.version_info)
```

```
atomickitty.aws
compthink
/opt/conda/envs/python/bin/python
3.8.3 (default, Jul 2 2020, 16:21:59)
[GCC 7.3.0]
sys.version_info(major=3, minor=8, micro=3, releaselevel='final', serial=0)
```

```
[2]: %%html
<!--Script block to left align Markdown Tables-->
<style>
  table {margin-left: 0 !important;}
</style>
```

```
[2]: <IPython.core.display.HTML object>
```

1 ENGR 1330 Computational Thinking with Data Science

1.1 Course Description:

Introduction to Python programming, its relevant modules and libraries, and computational thinking for solving problems in Data Science. Data science approaches for importing, manipulating, and analyzing data. Modeling and visualizing real-world data sets in various science and engineering disciplines.

3 credit hours comprising of lectures and hands-on lab sessions.

This course provides a hands-on learning experience in programming and data science using iPython and JupyterLab. iPython is the interactive python kernel implemented in JupyterLab.

1.2 Prerequisites:

Prior programming background is NOT required. The course is intended for first-year WCOE students (aka engineering foundational)

1.3 Course Sections (in this syllabus)

Lesson time, days, and location:

1. Section 003; CRN 42882; 0800-0920 T, TH ; Telepresence
 1. Lab Section D52; CRN 43065; 0930-1050 T, TH
2. Section 004; CRN 42884; 0800-0920 T, TH ; Telepresence
 2. Lab Section D54; CRN 43067; 0930-1050 T, TH

1.4 Course Instructor:

Instructor: Theodore G. Cleveland, Ph.D., P.E., M. ASCE, F. EWRI

Email: theodore.cleveland@ttu.edu (put ENGR 1330 in subject line for email related to this class, identify your section 003 or 004)

Office location: Telepresence (Zoom; GoToMeeting; etc.)

Office hours: TBD

1.5 Teaching assistants:

Assistant: Farhang Forghanparast

Email : Farhang.Forghanparast@ttu.edu (put ENGR 1330 in subject line for email related to this class, identify your section 003 or 004)

Office location: Telepresence

Office hours: MW 0900-1000 (Zoom Meeting/BB Collaborate)

1.6 COVID-19 Important Guidelines:

- If Texas Tech University campus operations are required to change because of health concerns related to the COVID-19 pandemic, it is possible that this course will move to a fully online delivery format. Should that be necessary, students will be advised of technical and/or equipment requirements, including remote proctoring software.
- Policy on absences resulting from illness: We anticipate that some students may have extended absences. To avoid students feeling compelled to attend in-person class periods when having symptoms or feeling unwell, a standard policy is provided that holds students harmless for illness-related absences (see Section A below).

1.6.1 A. Illness-Based Absence Policy (Face-to-Face CLasses)

If at any time during the semester you are ill, in the interest of your own health and safety as well as the health and safety of your instructors and classmates, you are encouraged not to attend face-to-face class meetings or events. Please review the steps outlined below that you should follow to ensure your absence for illness will be excused. These steps also apply to not participating in

synchronous online class meetings if you feel too ill to do so and missing specified assignment due dates in asynchronous online classes because of illness.

1. If you are ill and think the symptoms might be COVID-19-related:
 1. Call Student Health Services at 806.743.2848 or your health care provider. During after-hours and on weekends, contact TTU COVID-19 Helpline at TBD.
 2. Self-report as soon as possible using the Dean of Students COVID-19 webpage. This website has specific directions about how to upload documentation from a medical provider and what will happen if your illness renders you unable to participate in classes for more than one week.
 3. If your illness is determined to be COVID-19-related, all remaining documentation and communication will be handled through the Office of the Dean of Students, including notification of your instructors of the time you may be absent from and may return to classes.
 4. If your illness is determined not to be COVID-19-related, please follow steps 2.a-d below.
2. If you are ill and can attribute your symptoms to something other than COVID-19:
 1. If your illness renders you unable to attend face-to-face classes, participate in synchronous online classes, or miss specified assignment due dates in asynchronous online classes, you are encouraged to contact either Student Health Services at 806.743.2848 or your health care provider. Note that Student Health Services and your own and other health care providers may arrange virtual visits.
 2. During the health provider visit, request a “return to school” note.
 3. E-mail the instructor a picture of that note.
 4. Return to class by the next class period after the date indicated on your note.

Following the steps outlined above helps to keep your instructors informed about your absences and ensures your absence or missing an assignment due date because of illness will be marked excused. You will still be responsible to complete within a week of returning to class any assignments, quizzes, or exams you miss because of illness.

1.6.2 B. Illness-Based Absence Policy (Telepresence CLasses)

Same as above with respect potential to infect others; go to a health care provider if you are ill. Telepresence courses are recorded and will be available on TTU MediaSite and/or YouTube (unlisted). Exercises, Quizzes, and Examinations are all administered by a Learning Management System (Blackboard) and users need to allow enough time to complete and upload their work>

1.7 Textbook:

Ani Adhikari and John DeNero, Computational and Inferential Thinking, The Foundations of Data Science, Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0). Link: <https://www.inferentialthinking.com/chapters/intro>.

1.8 Course Contents:

- Computational thinking for problem-solving: Logical problem solving, decomposition, pattern recognition, abstraction, representation, algorithm design, and generalization.
- Python Programming:

1. Variables, constants, data types, data structures, strings, math operators
 2. boolean operators, expressions, program constructs, functions,
 3. looping, I/O files, modules, and database.
- Data science fundamentals:
 1. Experimental setup:
 1. Importing and formatting data sets,
 2. Displaying data,
 3. Data pre-processing.
 2. Introductory statistical analysis with Python:
 1. Elementary statistics, randomness, sampling, probability distributions,
 2. Confidence intervals, hypothesis testing, and A/B testing.
 3. Basic data analysis, visualization, and machine learning:
 1. Data pre-processing,
 2. Supervised/unsupervised learning,
 3. Performance evaluation metrics.

1.9 Learning Outcomes:

On completion of the course, students will have * Created Python programs employing computational thinking concepts to * Employed Python libraries relevant to data science. * Downloaded data files from selected public sources and analyzed content. * Created scripts to perform fundamental data analytics and basic visualization.

1.10 ABET Student Outcomes

- Engineering:
 1. An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics.
 2. An ability to acquire and apply new knowledge as needed, using appropriate learning strategies.
- Computer Science:
 1. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
 2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

1.11 Resources/Tools

1.11.1 Platforms for Python Programming (for your own computers)

1. Anaconda platform (<https://www.anaconda.com/>): Anaconda distribution is an open-source Data Science Distribution Development Platform. It includes Python 3 with over 1,500 data science packages making it easy to manage libraries and dependencies. Available in Linux, Windows, and Mac OS X.
2. Jupyter (<https://jupyter.org/>): JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data. JupyterLab is flexible: Configure and arrange

the user interface to support a wide range of workflows in data science, scientific computing, and machine learning. **note** Anaconda for MacOS includes a JupyterLab instance, so a separate install is not required.

1.11.2 Additional Modules for Python Programming

3. Math module (<https://docs.python.org/3/library/math.html>): Gives access to the mathematical functions defined by the C standard e.g. factorial, gcd, exponential, logarithm.
4. Operator module (<https://docs.python.org/3/library/operator.html>): Helps in exporting a set of efficient functions corresponding to the intrinsic operators of Python. For example, the operator `add(x,y)` is equivalent to the expression `x+y`.

1.11.3 Python Modules for Data Science

5. Scipy module (<https://www.scipy.org/>): A Python-based ecosystem of open-source software for mathematics, science, and engineering. Some of the core packages are:
 - Numpy: Provides n-dimensional array package
 - Scipy: Fundamental for scientific computing (e.g. linear algorithm, optimization)
 - Matplotlib: Visualizations/2D plotting
 - IPython: Enhanced interactive console «= this is the kernel used in JupyterLab
 - Pandas: Data structures and data analysis
6. Scikit-learn module (<https://scikit-learn.org/stable/>): A library for machine learning in Python. It is a simple and efficient tool for predictive data analysis. It is built on NumPy, SciPy, and matplotlib modules.

1.11.4 Some thoughts on “your own machine”

If you already have an Apple computer: The MacOS install is the easiest, and does a good job of installing everything - with a caveat! MacOS uses python 2.7+ as part of the OS (for upgrades and such), you have to be sure you are using the correct environment: `conda activate/deactivate` will become your friend. Do not buy an Apple computer for this course or any engineering course. Apple makes a fine product, but professional engineering software is PC-centric. Apple computers using a hypervisor are fine, but that’s getting into complexity that will degrade the educational experience. As a prediction, the first few years of Apple on ARM chips is going to be a disaster (for Apple) as most x86-64 application layers do not play well with ARM when the function calls are recursive – but Apple will eventually work it out.

My advice for WCOE students is to buy a PC if you can, even a cheap POS will serve you well at TTU.

A Windows 10 install is almost as easy, be sure you accept default paths of you will never get it to work right.

A Linux install is the hardest of the three because of environment settings. I found Ubuntu (Debian based, so same for Raspian) easier than CentOS (RHEL based). Some things to remember, don’t run conda as `sudo` it puts stuff in wrong place.

On-Line Options While network dependent the <https://cocalc.com/> environment is a good option, they have the kernels installed and running, you log-in and work on their servers. The free tier is probably more than adequate for this class.

1.11.5 CECE Support Site

The instructor maintains a web-site with a lot of useful content related to this course; nearly all the examples I will use in class will be stored at this site in addition to TTU-owned sites. Feel free to visit and use the materials with attribution located at <http://atomickitty.ddns.net/documents/JupyterOrion/>

In fact this syllabus was created using a JupyterLab notebook (as a markdown processor).

1.12 Course Schedule

Item	Lesson	Lab
25 Aug 2020	1. Introduction to Computational Thinking with Data Science: - Computational thinking concepts - Python as a programming environment - Data science and practices	Environment set up: - Jupyter notebook - Computational Thinking Examples
27 Aug 2020	2 .Programming Principles: - Data types (int, float, string, bool) - Variables, operators, expressions, basic I/O - String functions and operations	Introduction to Python - Data types (e.g. int, float, string, bool) - Expressions
1 Sep 2020	3. Programming Principles: - Data structures: Array, list, tuple, set, dictionary - Conditional statements - Loops	Introduction to Python - Data structures - Conditional statements - Loops
3 Sep 2020	4 . Program flow control (Loops) - Controlled repetition - Increment skip if greater - Decrement skip if equal	Introduction to Python - Structured FOR Loop - Structured WHILE Loop
8 Sep 2020	5 .Programming Principles: - Functions - Variable scope	Introduction to Python - Functions - Variable scope
10 Sep 2020	6 .Programming Principles: - Class and objects - File handling	Files - Reading from a file - Writing to a file - Get file from URL
15 Sep 2020	7. Data Representation and Operations: Python library: NumPy - Data representation: Arrays, vectors, matrices - Data operations: Indexing, math functions	Exercises on NumPy

Item	Lesson	Lab
15 Sep 2020	8. Data Query and Manipulation: Python Library: Pandas - Data frame: Create, index, read/write to file, summarize statistics, and fill and drop values	Exercises on Pandas
17 Sep 2020	9. Data Display: Python Libraries: Matplotlib - Data Display for line charts, bar charts, box plot, scatter plot, and histograms	Exercises on data display
22 Sep 2020	Review – NumPy, Pandas, Matplotlib	Midterm-1 - LMS administered
24 Sep 2020	Data Modeling: Statistical Approach: - Establishing causality - Randomness: Iteration, simulation	Exercises on causality and simulation
29 Sep 2020	Randomness: Probabilities	Exercises on probabilities
1 Oct 2020	Sampling and empirical distributions	Exercises on sampling
1 Oct 2020	Hypothesis testing: General concept and examples of assessing models.	Exercises on hypothesis testing
6 Oct 2020	Hypothesis testing: Comparing proportions, type1 & type2 errors, p-value.	Exercises on hypothesis testing
8 Oct 2020	Comparing two samples: A/B Testing	Exercises on A/B testing
13 Oct 2020	Confidence intervals	Exercises on confidence intervals
15 Oct 2020	Center and spread	Exercises on center and spread
20 Oct 2020	Normal distribution	Exercises on normal distribution
22 Oct 2020	Sample means	Exercises on sample means
27 Oct 2020	Probability estimation modeling	Exercises
29 Oct 2020	Review – Statistical analysis	Midterm-2 - LMS administered
3 Nov 2020	Data Modeling: Machine Learning Approach: - Correlation - Presentation template - Final projects	Exercises on correlation

Item	Lesson	Lab
5 Nov 2020	Ordinary least squares (OLS) regression	Exercises on least squares
10 Nov 2020	Residuals	Exercises on computing residuals
12 Nov 2020	Regression Inference	Exercises on regression
17 Nov 2020	Regression: - Evaluation metrics: Accuracy, error	Exercises on regression with evaluation
19 Nov 2020	Classification: - Supervised learning- Nearest neighbor	Exercises on KNN
24 Nov 2020	Classification Evaluation and Making Decisions: - Confusion matrix, precision, recall, accuracy, F-score. - Making decisions	Exercises on KNN with evaluation
2 Dec 2020	Review – Machine learning	Review - All Content
9 Dec 2020	Final Exam - LMS administered	

1.13 Course Assessment and Grading Criteria:

There will be three midterm exams and one comprehensive final project for the course. In addition, lab participation, quizzes, and assignments also contribute to the final grade. Late assignments will not be scored.

Grades will be based on the following components; weighting is approximate:

Assessment Instrument	Total points	Weight(%)
Midterm-1	100	16
Midterm-2	100	16
Midterm-3	100	16
Lab participation	30	5
Quizzes	100	16
Assignments	70	11
Final project	100	16
Overall total	600	100

Letter grades will be assigned using the following proportions:

Normalized Score Range	Letter Grade
90	A
80-89	B
70-79	C
55-69	D
< 55	F

1.14 Classroom Policy:

The following activities are not allowed in the classroom: Texting or talking on the cellphone or other electronic devices, and reading non-course related materials. ### Telepresence (On-line) Courses Obviously electronic devices are vital; disrupting the webinar is prohibited, please mute your microphone unless you have a question - consider typing your question into the chat window as well.

1.15 ADA Statement:

Any student who, because of a disability, may require special arrangements in order to meet the course requirements should contact the instructor as soon as possible to make necessary arrangements. Students must present appropriate verification from Student Disability Services during the instructor's office hours. Please note that instructors are not allowed to provide classroom accommodation to a student until appropriate verification from Student Disability Services has been provided. For additional information, please contact Student Disability Services office in 335 West Hall or call 806.742.2405.

1.16 Academic Integrity Statement:

Academic integrity is taking responsibility for one's own class and/or course work, being individually accountable, and demonstrating intellectual honesty and ethical behavior. Academic integrity is a personal choice to abide by the standards of intellectual honesty and responsibility. Because education is a shared effort to achieve learning through the exchange of ideas, students, faculty, and staff have the collective responsibility to build mutual trust and respect. Ethical behavior and independent thought are essential for the highest level of academic achievement, which then must be measured. Academic achievement includes scholarship, teaching, and learning, all of which are shared endeavors. Grades are a device used to quantify the successful accumulation of knowledge through learning. Adhering to the standards of academic integrity ensures grades are earned honestly. Academic integrity is the foundation upon which students, faculty, and staff build their educational and professional careers. [Texas Tech University ("University") Quality Enhancement Plan, Academic Integrity Task Force, 2010].

1.17 Religious Holy Day Statement:

"Religious holy day" means a holy day observed by a religion whose places of worship are exempt from property taxation under Texas Tax Code §11.20. A student who intends to observe a religious holy day should make that intention known to the instructor prior to the absence. A student who is absent from classes for the observance of a religious holy day shall be allowed to take an examination or complete an assignment scheduled for that day within a reasonable time after the absence. A student who is excused may not be penalized for the absence; however, the instructor may respond appropriately if the student fails to complete the assignment satisfactorily.

1.18 Ethical Conduct Policy:

Cheating is prohibited, and the representation of the work of another person as your own will be grounds for receiving a failing grade in the course.

[0]: