THE UNIVERSITY OF TEXAS AT EL PASO
Woody L. Hunt College of Business
Department of Accounting and Information Systems

Eduardo D. Villacis-Calderon
Office Room: CoBA 216
Phone: (915) 747-6028; Email: edvillacisc@utep.edu

CIS 3301 – Intro to Data Process & Prog

Course Syllabus, SPRING 2023

Class Hours: Mondays and Wednesdays 1:30 – 2:50 PM at CoBA 302
Office Hours: Mondays and Wednesdays 3:00 – 6:00 PM (or by appointment)

Course Description

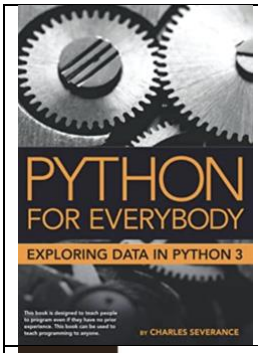
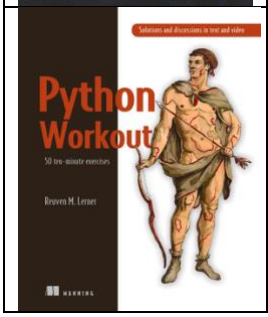
This is an introductory course in programming and data processing. This course teaches important concepts about programming blocks, algorithms, and data structures using python. More than teaching you how to program in python, this course teaches programming fundamentals that can be used to implement code in python or any other programming language. Moreover, this course teaches how to design, implement, and evaluate programming solutions to business problems. Furthermore, this course offers an introduction to the processing of data in multiple formats (e.g., plain files, comma separated values). This is a programming course. Therefore, in this course, coding, debugging, and improving code are central to students' learning.

Learning Objectives

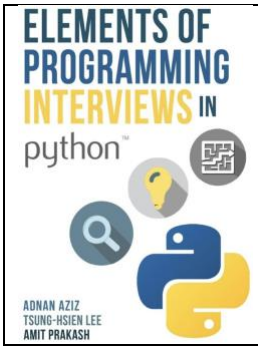
Upon successful completion of this course, students will be able to:

- Understand and utilize basic programming building blocks
- Analyze algorithms implemented in code
- Solve problems by thinking algorithmically and implementing code
- Read and write files with different formats
- Clean and transform data from files
- Understand concepts of object oriented programming
- Build and instantiate objects
- Handle errors and exceptions

Required books

	Severance, Charles R. Python for everybody. Charles Severance, 2009. Available for free at the author's website. https://www.py4e.com/book
	Lerner, R. (2020). Python workout: 50 ten-minute exercises (1st edition). Manning. Available for free at UTEP's library. https://utep.primo.exlibrisgroup.com/permalink/01UTEP_INST/1q3tr5t/cdi_askewsholts_vlebooks_9781638357223

Optional books

	Aziz, A., Lee, T. H., & Prakash, A. (2019). Elements of programming interviews in Python. EPI.

Required software

python	Python 3.0 or greater. https://www.python.org/
	Visual Studio Code. https://code.visualstudio.com/
	Git. https://git-scm.com/book/en/v2/Getting-Started-Installing-Git
	GitHub Desktop. https://desktop.github.com/

Teaching methods

1. Class assignments (CASA)
    i. It is important for students to come prepared to class and engage in each session. Therefore, before each lecture, students will be assigned a CASA assignment. These assignments have the goal of testing the preparation that students have before each lecture. All CASAs are due one hour before the lecture. Only students that attend in person or present a formal justification for their absence will receive credit for CASA submissions.
2. Coding assignments (CODE)
    i. Coding is essential for this class. Therefore, students will be assigned a coding assignment after each class session. Coding assignments will only be available 48 hours after each class session, and they will be evaluated in full grade if they run without errors. Coding assignments that do not run or run with errors will automatically receive a 50% grade penalty.
    ii. Students are allowed to consult their notes, books, classmates, or the internet to complete coding assignments. However, copying code blindly, or worst, plagiarizing code from others, is prohibited. Students should be able to explain how their code works and why code similarities appear. Students that fail to explain and justify their code authorship will receive a zero on the assignment and will be reported to the Honor Code System.
3. Exams (EXAM)
    i. Three will be three learning assessments. A review session will be held before each exam. It is recommended that you review the exam material before each review session so you can actively participate in review sessions. Exams in this class are comprehensive and will require students to code.

Evaluation

Class assignments – 10 points for each assignment (24 CASA assignments)
Coding assignments – 20 points for each discussion (23 CODE assignments)
Exams – 100 points (3 Exams)

A	B	C	D	F
900	800	700	600	<600

Late assignments
Late assignments will be awarded 20% less credit per day late. Make-up assignments, class assignments, discussions, checkpoints, and presentations will not be given.

Extra credit
The course incorporates extra credit opportunities in assignments, in-class activities, and discussions to promote students' engagement inside and outside the classroom.

Email Procedure
Please include "CIS 3301" in the subject line of all emails to the instructor to ensure that they are properly filtered. It would be helpful if the subject line also included a brief statement of need, for example: "CIS 3301 – Request for Appointment." Please read the following link about emailing a professor for some helpful suggestions (e.g., please start with a greeting including my name and a signature including your name): http://www.wikihow.com/Email-a-Professor.

Accommodations
If you need special accommodations due to a disability, as recognized by the Americans with Disabilities Act, please contact The Center for Accommodations and Support Services (CASS) at 747-5148 or email at cass@utep.edu, or visit their office located at UTEP Union East, Room 106. For additional information, please visit the CASS website at www.sa.utep.edu/cass.

Academic integrity
Academic integrity is an extremely serious matter. All students are expected to comply with University rules and regulations on academic integrity and honesty. Disciplinary sanctions may be imposed for violations of these rules and regulations. If you have questions or are unclear about what constitutes academic misconduct on an assignment, please speak with me. I take the honor code very seriously in the course.

***Tentative* Course Outline**

Module 1: Programming building blocks	
Week #1	
01/18 - Session 0	Introduction to CIS 3301
Week #2	
01/23 – Session 1	Computer architecture and programming building blocks
01/25 – Session 2	Basics of python semantic and syntax
Week #3	
01/30 – Session 3	Variables, statements, operators, and expressions
02/01 – Session 4	Basic data types (Integer, float, string) and data structures (lists, tuples)
Week #4	
02/06 – Session 5	Arithmetic and string basic operations
02/08 – Session 6	Conditional code execution (if, match, while)
Week #5	
02/13 – Session 7	Iteration code execution (for)
02/15 – Session 8	Built-in, type conversion, math, and customized functions
Module 2: Algorithms and data structures	
Week #6	
02/20 – Session 9	Exam 1 - Review
02/22 – Session 10	Analysis of algorithms (Big-O)
Week #7	
02/27 – Session 11	Data structures (lists, tuples)
03/01 – Session 12	Data structures (sets, dictionaries)

Week #8	
03/06 – Session 13	Search algorithms (linear, binary)
03/08 – Session 14	Sorting algorithms (bubble, selection)
Week #9	
03/13 – Session 15	Spring break
03/15 – Session 16	Spring break
Week #10	
03/20 – Session 17	Recursion
03/22 – Session 18	Sorting algorithms with recursion (bubble, merge)
Module 3: Data Processing	
Week #11	
03/27 – Session 19	Exam 2 - Review
03/29 – Session 20	Text encodings (ASCII, UTF-8, Unicode) and regular expressions
Week #12	
04/03 – Session 21	Read and write plain text files
04/05 – Session 22	Read and write comma separated files
Week #13	
04/10 – Session 23	Read and write comma separated files using Pandas
04/12 – Session 24	Data cleaning and transformation using Pandas
Module 4: Object oriented programming	
Week #14	
04/17 – Session 25	Introduction to the object oriented programing paradigm
04/19 – Session 26	Create and instantiate objects
Week #15	
04/24 – Session 27	Inheritance and objects life cycle
04/26 – Session 26	Error and exceptions handling
Week # 16	
05/01 – Session 28	Web scrapping using http request and beautiful soup
05/03 – Session 29	Exam review
Final Exam	

Important Notes:

1. In addition to the announced office hours, students may stop by my office at any time (or email me) to ask questions.
2. Students should demonstrate that they have done code troubleshooting/debugging before requesting assistance from the instructor.
3. If you have any trouble with the class, please get help ASAP. Do not let the problems build up.
4. This syllabus is tentative.