Python for Data Science I

DATA 3401

Spring 2024

Instructor: Max (Masoud) Rostami
Office: Life science building,
Room: Life Science 231
e-mail: masoud.rostami@uta.edu
Office hours: 2:30 PM-3:30 PM, MW

Class start/end: January 17, 2024 – April 29, 2024
Lecture meeting times: MW 1:00 PM-2:20 PM
Lecture meeting place: SH 330 / Microsoft Teams
Lab meeting times: Friday 3:00 PM-4:50 PM
Lab meeting place: PKH 111 / Microsoft Teams as needed

Teaching Assistants:	Sarath Chintakunta, Sridurga Linga
office:	Virtual via Teams
e-mail:	Sarath Chintakunta (sxc6989@mavs.uta.edu); Sridurga Linga (sxl6081@mavs.uta.edu)
office hours:	Sarath Chintakunta (Tuesday 10am-1 pm) Sridurga Linga (Monday 10 am-12 pm)

COVID-19 Notes:

All classes will be held in person (or other modalities if needed). Please avoid attending the lectures in person if you do not feel well for any reason. The instructor will ensure students have access to all materials, homework, and quizzes online if they cannot attend the lectures in person due to sickness or other reasons. PLEASE TAKE YOUR HEALTH AND OTHERS IN THIS CLASS SERIOUSLY BY PAYING ATTENTION TO COVID19 PROTOCOLS. FAILURE TO DO SO COULD COST YOUR FRIENDS, CLASSMATES, OR YOUR INSTRUCTOR'S LIVES. Wearing a mask and vaccinating is not mandatory.

COURSE OBJECTIVES / ACADEMIC LEARNING GOALS

This is the first of a two-course sequence providing the necessary foundations in scientific computing for Data Science majors. It introduces a number of operating systems, languages, and tools using examples and contexts from natural and behavioral sciences. **Prerequisite**: None.

The primary objective of this course is to learn basic computer programming concepts and apply them to Data Science related problems. By the end of the course, you should have a good understanding of general programming foundations and practices and be able to analyze data-intensive problems and develop computational solutions for them, collaboratively within a team.

We will achieve this by learning how to program in popular operating system environments such as Linux using popular high-level programming languages such as Python. Although not essential, some prior level of familiarity with programming concepts is desirable for this course.

COURSE SCHEDULE

The following is a tentative outline of topics to be covered:

Week 1: Introduction to Programming History and Operating Systems

- How Computers Work
- The Importance of Programming
- History of Programming
- Introduction to Unix Shell
- Basics of Version Control Systems

Week 2: Unix shell and Version Control Systems and Git

- Principles of professional project management
- Collaborative programming with the use of Git
- Creating and managing repositories
- **Lab 1**
- **Homework 1**

Week 3: Python Programming

- Python versus other programming languages
- Installing Python on different platforms
- Basic syntax rules
- Naming and Using Variables
- Explore Python Syntax
- Variable types and operators

- **Lab 2**
- **Homework 2**

Week 4: Conditional Statements, Loops

- Comparisons and logical operators
- Conditional Statements
- Checking for Equality and Inequality
- Checking Multiple Conditions
- While loops
- For loops
- Nested loops
- Break, Continue
- **Lab 3**
- **Homework 3**

Week 5: Functions and Modules

- Introduction to functions
- Variable Scope
- Defining and calling functions
- Arguments and parameters
- Built-in functions and modules
- Lambda Functions and Functional Programming Concepts
- **Lab 4**
- **Homework 4**

Week 6: Deep Dive into Data Structures

- Exploring Lists, Tuples, Dictionaries, and Sets
- Compound Data Structures and Indexing
- List Comprehensions
- Introduction to Iterators and Generators
- **Mid-Semester Project**

Week 7: Object Oriented programming (OOP) in Python

- Introduction to OOP Concepts
- Classes, Objects, Attributes, and Methods
- Principles of Object-Oriented Design
- Instance Creation from Classes
- **Lab 5**
- **Homework 5**

Week 8: Advance Object Oriented programming

- Inheritance, Polymorphism, and Encapsulation
- Creating and Managing Subclasses
- Overriding Methods and Defining Child Class Attributes
- Importing Classes in Python
- **Lab 6**
- **Homework 6**

Week 9: Spring Break

- A break for students to rest and review past materials.

Week 10:Error Handling, File I/O, Automation

- Basic File Handling with Python
- Understanding Exceptions and Errors.
- Techniques for Reading and Writing Files (Text, JSON, CSV)Fundamentals of Python Automation
- Script Writing for Automating Simple Tasks
- Advanced File Handling (Text, CSV)
- Projects on Python Automation
- Creating an Email Autoresponder with Python
- **Lab 7**
- **Homework 7**

Week 11: Data Science Libraries: NumPy and Random Variables

- Introduction to NumPy

- Creating and manipulating arrays
- Random Variables
- Array indexing and slicing.
- Mathematical operations and statistical functions
- Working with Multi-dimensional Arrays
- Advanced Array Indexing
- Working with images
- SciPy
- **Introducing the students to the Group Presentation subjects and Expectations**
- **Lab 8**
- **Homework 8**

Week 12: Data Science Libraries: Pandas

- About Data Frames
- Creating Data Frames
- Introduction to Pandas
- Interacting with Data Frame Data
- Reading and writing data to and from different file formats
- Creating and manipulating Data Frames
- **Lab 9**
- **Homework 9**

Week 13: Data Science Libraries: Pandas, Data Wrangling

- Data wrangling and cleaning
- Grouping and aggregating data
- Data Manipulation: Using Loops and Data Structures to Process Data
- **Lab 10**
- **Homework 10**

Week 14: Data Visualization

- Introduction to Visualization
- Good and Bad Graphs
- Introduction to Matplotlib
- Creating and customizing plots

- Line plots, bar plots, scatter plots, and histograms
- Heatmaps and contour plots
- **Lab 11**
- **Homework 11**

Week 15: Group Presentation

- Students will present their subject of interest as a group. All the presentations will be in-person.

Week 16: Group Presentation

- Some of the presentations may schedule to take place virtually on Friday, May 3, from 2:00 to 5 p.m If we need an extra day.

COPYRIGHT POLICY:

Any material that I produce throughout the course is copyrighted, including (but not limited to) syllabi, quizzes, worksheets, exams, reviews, or any in-class material.

DISCLAIMER:

The information here is subject to change at my discretion for the purpose of better facilitating student learning and success.

COURSE TEXTBOOKS

No textbook is required for this course. Online class lecture notes will be used as reference. However, a list of textbooks for those who are interested to self-educate themselves or go beyond class syllabus will be provided on the first day of the class.

GRADING:

Weekly Homework: 45% (Assignments might not be weighted equally)

Weekly Labs: 25%

Midterm Exam: 10%

Project & Presentation: 15%

Attendance: 5%

Homework Policy:

There will be approximately one homework per week. Assignments will be due before the lecture begins on Friday of the following week and should be added to an online repository determined by the

instructor. No late assignments will be accepted. No exceptions to the homework policy will be made without prior instructor approval.

Attendance:

Attendance is an essential part of this course. Students are expected to attend all classes and arrive on time. To ensure that students are keeping up with the course material, each student will be allowed to just have up to five absences during the semester without penalty. For every absence beyond five, one percent will be deducted from the total attendance grade. If a student misses class due to an illness or emergency, they should notify the instructor as soon as possible and provide appropriate documentation such as compelling evidence of illness or an official letter from the university administration. Excused absences will not count against the five allowed absences.

Please note that attendance is not only for the purpose of receiving credit but also for the purpose of gaining knowledge and engaging in the learning process. Being present in class ensures that students have access to important information, discussions, and activities that contribute to their overall understanding of the course material.

Scholastic dishonesty:

All students are responsible for upholding the University rules on scholastic dishonesty. Students who violate University rules on scholastic dishonesty are subject to disciplinary penalties, including the possibility of failure in the course and/or dismissal from the University. Since such dishonesty harms the individual, all students, and the integrity of the University, policies on scholastic dishonesty will be strictly enforced.

Other matters: The University of Texas at Arlington provides, upon request, appropriate academic adjustments for qualified students with disabilities. Any student with a documented disability (physical or cognitive) who requires academic accommodations should contact the UTA's Office for Students with Disabilities as soon as possible to request an official letter outlining authorized accommodations. For visit https://www.uta.edu/disability/.

Your Expectations:

For the current offering of this course, we will cover the principles of computer programming using the Python programming language. Specifically, upon completion of this course you will be familiar with,

- programming paradigms,
- principles of software maintenance and collaborative project development,
- differences between compiled and interpreted programming languages,

- how to use Python as a simple calculator,
- how to use Python as an advanced scientific computation and graphics toolbox,
- how to use Python to collect, clean, and analyze data and make a scientific inference,
- how to formulate and cast a scientific problem in the form of a sequence of computational algorithms.