# IST356: Programming Techniques for Data Analytics

## Course Description

The exploration of common approaches for building data pipelines used in analytics. Emphasis on both exploratory and production scenarios. Programming experience required.

### Additional Course Description

This course is a tour of programming techniques for building data pipelines for analytics. It will not just emphasize exploratory approaches, but also techniques to build extract transform load pipelines to run code in production. Throughout the course we will learn how to source data from a variety of sources (files, data streams, APIs, web scraping, etc.) and ultimately transform data as to prepare it for dashboards or machine learning. You will also learn some simple data visualization but that is not the primary emphasis of the course.

### Prerequisites

This courses uses the Python programming language. Proficiency n any programming language is the only pre-requisite. Students should have a clear understanding of these concepts:

* Input, output, variables and data types
* Control flow statements (if, for, while)
* Functions (function definition, calling, parameters, return values)
* Data structures (lists, dictionaries)
* Using code in other libraries

### Audience: IST256 or IST356?

This course is intended as a follow up course to IST256. It is also appropriate for students with prior experience with programming who have an interest in data analytics.

* IST256 is for students with little to no programming experience. The course content is 75% python fundamentals and 25% python for data analytics.
* IST356 is for students with prior programming experience. The course content is 25% python fundamentals and 75% python for data analytics.

### Credits

3 credits

### Course Fees

None

## Learning Objectives

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

1. Explain techniques for sourcing or transforming data, and be able to justify the choice of technique
2. Solve data-oriented problems using programming techniques
3. Evaluate different code modules and application programming interfaces for suitability
4. Apply data transformational programming techniques to build a larger data pipelines
5. Create production quality data pipelines from exploratory code

## Textbooks And Supplies

### Textbooks

There are no textbooks for this course. All required readings are available freely online. The online sources are the most up-to-date references to the relevant course material.

#### Python

* PY4E - Python for Everybody. (2018). Py4e.com. <https://www.py4e.com/book>
* The Python Language Reference. (2022). Python Documentation. <https://docs.python.org/3.11/reference/index.html>
* The Python Standard Library. (2022.). Python Documentation. <https://docs.python.org/3.11/library/index.html>

#### User Interface

* Widget List — Jupyter Widgets 8.1.0 documentation. (2022). Readthedocs.io. <https://ipywidgets.readthedocs.io/en/latest/examples/Widget%20List.html>
* Using Interact — Jupyter Widgets 8.1.0 documentation. (2017). Readthedocs.io. <https://ipywidgets.readthedocs.io/en/latest/examples/Using%20Interact.html#using-interact>
* API Reference - Streamlit Docs. (2023). Docs.streamlit.io. <https://docs.streamlit.io/library/api-reference>

#### Data Wrangling

* Stepanek, H. (2020). Thinking in Pandas : how to use the Python data analysis library the right way. Apress. ISBN: 9781484258392 (Available online via SU Library)
* API reference — pandas 2.0 documentation. (2023). Pandas.pydata.org. <https://pandas.pydata.org/docs/reference/index.html>
* API reference — Polars documentation. (2023). Pola-Rs.github.io. <https://pola-rs.github.io/polars/py-polars/html/reference/>

#### Web API’s

* Developer Interface — Requests 2.31.0 documentation. (2023). Requests.readthedocs.io. <https://requests.readthedocs.io/en/latest/api/> ‌- Fudge, M. (2018) Web APIs In Python. Ist256.com. <https://ist256.com/readings/Web-APIs-In-Python.html>
* OpenAPI Specification - Version 3.0.3 | Swagger. (2020). Swagger.io. <https://swagger.io/specification/>
* RFC6749. The OAuth 2.0 Authorization Framework (2012). Datatracker.ietf.org. <https://datatracker.ietf.org/doc/html/rfc6749> ‌- FastAPI. (2023). FastAPI. Fastapi.tiangolo.com. <https://fastapi.tiangolo.com/>

#### Data Visualizations

* API Reference — Matplotlib 3.5.0 documentation. (2023). Matplotlib.org. <https://matplotlib.org/stable/api/index>
* API reference — seaborn 0.11.2 documentation. (2023). Seaborn.pydata.org. <https://seaborn.pydata.org/api.html>
* Quickstart — Folium 0.12.1 documentation. (2023. Python-Visualization.github.io. <https://python-visualization.github.io/folium/quickstart.html>
* Plotly Express. (2023). Plotly.com. <https://plotly.com/python/plotly-express/>
* Plotly.(2023). Plotly Python Graphing Library. Plotly.com. <https://plotly.com/python/>

#### Web Scraping

* Real Python. (2018, January 23). Practical Introduction to Web Scraping in Python. Realpython.com; Real Python. <https://realpython.com/python-web-scraping-practical-introduction/>
* Richardson, L. (2019). Beautiful Soup Documentation — Beautiful Soup 4.4.0 documentation. Crummy.com. <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
* requests-HTML v0.3.4 documentation. (2023). Requests.readthedocs.io. <https://requests.readthedocs.io/projects/requests-html/en/latest/>
* HTML elements reference. (2019, June 6). MDN Web Docs. <https://developer.mozilla.org/en-US/docs/Web/HTML/Element>

### Jupyterhub Programming Environment

Our programming environment is a private-cloud web application called Jupyter Hub. https://v2hub.ischool.syr.edu. This is the de-facto programming environment of the scientific community. All students have an account; use your SU Microsoft Account (NetID and password) to login. After you login you will see a library folder inside that folder is an ist356 folder. All of the course content (lecture slides, code samples, assignments) is available in this folder.

## Course Requirements and Expectations

### Attendance and Participation

You are expected to attend and participate in every class session. If you must miss class, you are responsible for making up the work and catching up on what you missed. Do expect a pre-recorded lecture or instructor support if you miss class. A suggestion is to partner with a classmate to share notes from class.

### Assignments

The assignments are programming / problem solving activities that you will complete outside of class. The assignment due dates are posted on the course schedule. These assignment should be considered formative assessments. Use them to measure your knowledge of the course material to date. It is important to reflect upon your work and take an honest assessment of your abilities as you complete each assignment.

### Exams

Exams are summative assessments. They are designed to test your knowledge of the material. The exams will be issued in class on the dates posted on the course schedule. Exams are closed-book and will last about 45 minutes.

### Project

The project is your opportunity to demonstrate what you have learned in an experiential fashion. In your final project you will be expected to create a data pipeline of your choosing.

* The pipeline should incorporate techniques we learned in the course and the more techniques you incorporate correctly the higher your grade.
* It is expected that you will be able to explain your choices, and they will be appropriate for the problem.
* The pipeline should be first written exploratory, and then refactored into a production quality pipeline. Both pipelines should be submitted.
* There should be a simple data visualization or dashboard from the pipeline output as to demonstrate its usefulness.

## Grading

| **Assessment** | **Type** | **Learning  Outcomes** | **Quantity** | **Points Each** | **Points Total** |
| --- | --- | --- | --- | --- | --- |
| Assignments | Formative | 1, 2 | 10 | 2 | 30 |
| Exams | Summative | 3, 4 | 2 | 20 | 40 |
| Project | Summative | 5 | 1 | 30 | 30 |
| **Total** |  |  |  |  | **100** |

### Grading Table

The following grading scale translates your total points earned into a letter grade to be submitted to the University registrar.

| **Student Achievement** | **Total Points Earned** | **Registrar Grade** | **Grade Points** |
| --- | --- | --- | --- |
| Mastery | 475 - 500 | A | 4.000 |
|  | 450 - 474 | A- | 3.666 |
| Satisfactory | 425 - 449 | B+ | 3.333 |
|  | 400 - 424 | B | 3.000 |
|  | 375 - 399 | B- | 2.666 |
| Low Passing | 350 - 374 | C+ | 2.333 |
|  | 325 - 349 | C | 2.000 |
|  | 300 - 324 | C- | 1.666 |
| Unsatisfactory | 250 - 299 | D | 1.000 |
|  | 0 - 249 | F | 0.000 |

## Schedule Of Topics

### Topics

1. Python Essentials (Python)
   1. Input, output, variables, types, conditionals
   2. Iterations, lists, dictionaries
   3. Functions, strings, files
   4. Modules, import, pip
2. User Interface (UI)
   1. Ipywidgets and Interact in Jupyter
   2. Streamlit interactions
3. Data Wrangling
   1. Pandas, series, data frame, index, numpy types, Nan, slicing
   2. Loading / saving, file formats json, csv, parquet, pickle, HTML
   3. Filter, sort, project, json\_normalize, nested data, head, sample
   4. Joins, append, merge, lambdas, column generation, iteration
   5. Group by aggregations, pivot melt
   6. Polars vs Pandas
4. Web APIs
   1. HTTP basics, requests, caching, error handling
   2. Reading swagger, AWS, Azure, Google APIs
   3. Authentication, OAUTH2
   4. Hosting your own API with FastAPI
5. Data Visualization
   1. Plotting basics, matplotlib /seaborn
   2. Folium / Leaflet
   3. Plotly
   4. Choropleths
6. Web Scraping
   1. HTML basics, scraping techniques
   2. beautiful soup
   3. requests html
   4. Form submission, web driving

### Schedule

| **Week** | **Monday** | **Wednesday** | **Friday** |
| --- | --- | --- | --- |
| 1 | 8/28 | 8/30 | 9/1 |
|  | Course Introduction | Python 1-1 |  |
| 2 | 9/4 | 9/6 | 9/8 |
|  | **No Class: Labor Day** | Python 1-2 | Assignment 1 (A1) |
| 3 | 9/11 | 9/13 | 9/15 |
|  | Python 1-3 | Python 1-4 | Assignment 2 (A2) |
| 4 | 9/18 | 9/20 | 9/22 |
|  | UI 2.1 | UI 2.2 | Assignment 3 (A3) |
| 5 | 9/25 | 9/27 | 9/29 |
|  | Data Wrangling 3-1 | Data Wrangling 3-2 | Assignment 4 (A4) |
| 6 | 10/2 | 10/4 | 10/6 |
|  | Data Wrangling 3-3 | Data Wrangling 3-4 | Assignment 5 (A5) |
| 7 | 10/9 | 10/4 | 10/6 |
|  | **No Class: Fall Break** | Data Wrangling 3-5 |  |
| 8 | 10/16 | 10/18 | 10/20 |
|  | Data Wrangling 3-6 | Exam 1 (E1) | Assignment 6 (A6) |
| 9 | 10/23 | 10/25 | 10/27 |
|  | Web APIs 4-1 | Web APIs 4-2 | Assignment 7 (A7) |
| 10 | 10/30 | 11/1 | 11/3 |
|  | Web APIs 4-3 | Web APIs 4-4 | Assignment 8 (A8) |
| 11 | 11/6 | 11/8 | 11/10 |
|  | Data Visualization 5-1 | Data Visualization 5-2 | Assignment 7 (A7) |
| 12 | 11/13 | 11/15 | 11/17 |
|  | Data Visualization 5-3 | Data Visualization 5-4 | Assignment 9 (A9) |
| 13 | 11/20 | 11/22 | 11/24 |
|  | **No Class: Thanksgiving** | **No Class: Thanksgiving** |  |
| 14 | 11/27 | 11/29 | 12/1 |
|  | Web Scraping 6-1 | Web Scraping 6-2 | Assignment 8 (A8) |
| 15 | 12/4 | 12/6 | 12/8 |
|  | Web Scraping 6-3 | Web Scraping 6-4 | Assignment 10 (A10) |
| 16 | 12/11 | 12/20 | 12/22 |
|  | Exam 2 (E2) |  |  |
| 17 | 12/18 | 12/20 | 12/22 |
|  | Project Due (P) |  |  |