New York University Tandon School of Engineering
Dept. of Finance and Risk Engineering
FRE-GY6861 Financial Software Engineering Laboratory

Serge Feldman
Adjunct Professor

To contact professor: serge.feldman@nyu.edu
Office hours: by appointment

Course Summary

This half-semester course is of intermediate/advanced level, during which we'll take a deep dive into several advanced concepts of the Python ecosystem and explore development of large-scale real-life application using the language and other development tools.

Course Description

After a whirlwind review of language fundamentals, we will delve deeply into Python's advanced features, including data extraction and transformation, using table-like data representation (pandas dataframes). Furthermore, we will learn to employ the most widely used algorithms to filter, pivot, and aggregate data, using pandas library and run calculations, using numPy and sciPy libraries. At the same time, we will also learn to effectively use native built-in collection types, like tuple, lists, and dictionaries. Where needed, we will discuss other Python's powerful features, such as user-defined classes, object-oriented design, decorators, etc.

We will learn to apply industry-standard tools and techniques, such as GitHub (for code maintenance and review) and Atlassian Jira (for planning and tracking progress of our project), while working through implementation of a real-life system.

Project Description

We will implement a self-contained fully configurable Python-based workflow process, which allows to take a data source with variable format, extract data records and then standardize the target via multiple manipulation (pivot, aggregate, merge, map, etc.) and calculation operations. Finally, standardized dataframe is persisted into a data target. This infrastructure will have proper logging, error handling, and all other attributes of a real-life system.

More details will be provided during our meetings, during which we will have discussions of how to build various components.

Why Python and Data ETL Process

Python's status as the fastest-growing programming language is being fueled by a sharp uptick in its use for data science. This finding has been established by a new analysis by "Stack Overflow", the Q&A hub that is home to the world's largest online developer community. While Python is a versatile language with many data, ML, and math driven extensions (pandas, NumPy, SciPy, and a variety of AI tools), "Stack Overflow" found that one use case really stood out. Among visitors reading Python-tagged questions, there was a far greater rise in the proportion viewing questions related to data science, than those related to web development or systems administration.

On a related note, Python should be of particular interest to FRE students, because versions of Python-based quantitative and data manipulation infrastructures are running in Goldman Sachs (SecDB), J.P. Morgan (Athena), Bank of America (Quartz), etc.

Course Pre-requisites

Students will be expected to have solid grounding in at least one programming language, such as C#, or Java and should understand the concepts of functions, data structures and programming constructs of conditional and loop statements.

Ideally, students should already have fundamental knowledge of Python syntax.

This course is NOT suited for those that want to learn how to program and have no prior programming experience.

Required Text

None

Recommended Reading

Just about everything there is to know about Python can be found somewhere on the web by googling "python <name of feature>". Often, the answers can be found on stackoverflow.com or in the standard documentation maintained by the Python Software Foundation, docs.python.org, which is surprisingly readable.

During the course, students will be given multiple web-based articles, sample code, etc. to read, examine and review.

Technologies Used

As mentioned already, we will use Python 3 distribution with necessary libraries, GitHub, Jira, Jupyter and PyCharm IDEs. All of these tools are free to download and use.

We will discuss their installation, if necessary, during our first meeting.

Grading

Student's class participation - 25%
Student's participation in software development lifecycle - 25%
Development of already mentioned Python program – 50%

We will discuss all of that during our first meeting.

Detailed Course Outline

Note: Placement of topics in specific lectures is only approximate.

Lecture 1

1. Course Introduction
    - 1.1. Course "mechanics"

2. Technologies and tools to be used
    - 2.1. Python installation
    - 2.2. GitHub installation and usage
    - 2.3. Atlassian Jira usage
    - 2.4. Pycharm / Jupyter IDE installation and usage

3. Python Introduction
    - 3.1. What is Python
    - 3.2. What Python can do for you
    - 3.3. Primer of data types and variables
    - 3.4. Primer on conditions and loops
    - 3.5. Primer on functions
    - 3.6. Basics of objects
    - 3.7. Primer on modules

Lecture 2

4. Strings and Built-in Collections
    - 4.1. Strings manipulation
    - 4.2. Tuples
    - 4.3. Lists
    - 4.4. Dictionaries
    - 4.5. List comprehension

5. Project Introduction and Objectives
    - 5.1. High-level idea and workflow
    - 5.2. Scripts structure
    - 5.3. Main process skeleton

6. Project Implementation Inception
    - 6.1. Argument parsing
    - 6.2. Configuration with JSON format
    - 6.3. Logging
    - 6.4. Error handling

Lecture 3

7. Using *Pandas* to Explore Given Dataset

7.1. Environment setup
7.2. Getting to know your data
7.3. Getting to know *Pandas'* data structures (*Series* vs. dataframes)
7.4. Using indexing, *.loc* and *.iloc* operators
7.5. Data querying
7.6. Data grouping and aggregating
7.7. Columns manipulation
7.8. Cleaning data
7.9. Combining multiple datasets

8. Project Implementation (Cont.)
   8.1. File and directory access, using built-in *os.path* module
   8.2. Building an ETL pipeline, using *pandas*

Lecture 4

9. Project Implementation (Cont.)
   9.1. Building dataset aggregation operation
   9.2. Dynamic call of functions, using built-in *eval()* function
   9.3. Building dataset map and merge operation
   9.4. Manipulation of dataset columns (rename and reorder) implementation

Lecture 5

10. Memory Management
    10.1. Overview
    10.2. Objects in memory
    10.3. Garbage collection

11. Classes
    11.1. Overview
    11.2. Class attributes
    11.3. Class methods
    11.4. OOP inheritance introduction

Lecture 6

12. Project Implementation (Cont.)
    12.1. Testing and Running