2025 FRE Pre-Program Boot Camp Curricula

*Subject to minor changes

As of April 2025

The 2025 Boot Camp is designed as a 3-part series to ensure you absorb the information in a structured and logical sequence.

PART 3 - In-Person Course FRE-GY 5040

Dates: August 14, 2025 - August 29, 2025*

Instructor: Prof. Jason Yarmish

Time: Morning: 9:00AM - 12:00PM

Location: Tandon Campus, Room TBA

**Dates exclude August 27th & 28th, as New Student Graduate Orientation will take place on those dates*

Foundations of Applied Financial Engineering

Course Overview:

- This course equips students with practical fluency in the Applied Financial Engineering Lifecycle—from problem framing and data ingestion to risk-aware modeling and stakeholder communication. Through hands-on coding, homework, and a structured project, students learn about designing and implementing reproducible, stakeholder ready pipelines in Python. While the core focus is on the end-to-end data workflow, advanced topics such as deployment and orchestration are introduced for context, with resources provided for further study.

- As a foundational course in the program, it offers a comprehensive overview of the lifecycle. Later technical courses will plug into this framework with more details and deeper domain knowledge.

- Since not all topics can be covered in detail, throughout the syllabus we will use the following notation:
  - * Practically taught and used in the project
  - ~ Introduced conceptually
  - # Future topics with further study resources

What we will learn

- Define clear problem statements and scope
- Set up reproducible coding environments
- Build clean, reusable Python workflows for finance
- Pull and process real-world data from APIs and websites
- Clean, transform and explore datasets
- Apply regression, classification and time series models
- Use model performance metrics
- Draw conclusions while explicitly stating risks, assumptions, and design tradeoffs
- Communicate results transparently, with assumptions, risks, and user-facing framing for stakeholders
- Produce artifacts ready for stakeholders

Full Lifecycle Framework:

The work we do each day maps to stages of a real-world financial data pipeline:

- Problem Framing & Scoping
- Tooling Setup
- Python Fundamentals
- Data Acquisition/Ingestion
- Data Storage
- Data Preprocessing
- Outlier Analysis
- Exploratory Data Analysis ~
- Feature Engineering ~
- Modeling (Regression / Time Series / Classification)
- Evaluation & Risk Communication
- Results Reporting, Delivery Design & Stakeholder Communication ~
- Productization #
- Deployment & Monitoring #
- Orchestration & System Design #

Topics:

#	Date	Topics	Lifecycle Focus
1	8/14	Defining problem scope and framing, Python, NumPy, pandas, Notebooks, Envs, interpreted vs compiled languages	*Problem Framing & Scoping, *Tooling Setup, *Python Fundamentals
2	8/15	APIs, Web Scraping, Using .env files for API keys, Data Storage	*Data Acquisition/Ingestion, ~Data Storage
3	8/18	Cleaning, Filtering, Missing Data	*Data Preprocessing
4	8/19	Outliers, Assumption Sensitivity	*Outliers + Risk Assumptions
5	8/20	Visual & Statistical EDA	~Exploratory Data Analysis (EDA) ~Feature Engineering
6	8/21	Linear Regression + Assumptions	*Modeling: Linear Regression
7	8/22	Time Series + Classification Models	*Modeling: Time Series & Classification
8	8/25	Evaluation, Sensitivity, Communicating Risk with emphasis on uncertainty & assumptions in results	*Evaluation & Risk Communication
9	8/26	Lifecycle Review, Final Delivery, Stakeholder Communication, Production Concepts, Wrap-Up	*Results Reporting, Delivery Design & Stakeholder Communication, #Productization, #Deployment & Monitoring, #Orchestration, #System Design
10	8/29	Final Exam	

Pre-Class Setup:

- Homework 0 is to be completed before the first class and includes instructions for the following tasks:
    - Install VS Code and set up Python + conda environment
    - Run and submit a test Jupyter notebook
    - Upload and share via Google Classroom
    - Confirm access to course materials

Course Structure:

- Daily lectures and live code walkthroughs
- Step-by-step homework tied to each topic
- Ongoing applied project that builds daily
- Final in-class exam

Daily Homework Format:

Each day includes a structured coding task designed to:

- Reinforce lecture material through hands-on coding
- Build directly toward your final project deliverables
- Be submitted via Google Classroom

Project Details:

- Jupyter Notebook:
    - Clean, documented code with markdown annotations
- Slide Deck:
    - Stakeholder-ready presentation with charts, bullet-point conclusions, and an “Assumptions & Risks” section
- Code Repo:
    - Foldered project with README.md explaining workflow and lifecycle mapping
- Final Deliverable:
    - Clearly communicate results,
    - Explicitly states assumptions and risks,
    - Tailors format, framing and delivery to your audience

Final Exam:

- Topics & Depth
    - Cumulative multiple-choice test
- Style:
    - Mix of code comprehension, conceptual knowledge & lifecycle stages
    - “What happens if…” reasoning (sensitivity, risk)
- Grading:
    - Pass/Fail — students must demonstrate fluency with tools, concepts, and risk-aware thinking

Expectations:

Students are expected to attend class, participate in walkthroughs, and complete all daily assignments. Collaboration is encouraged, but final submissions must reflect individual understanding.

AI Use Policy:

AI tools may be used to check or gain understanding, but relying on them to complete your work undermines your learning and is not allowed. If you use AI to assist your thinking, briefly disclose how and where. This course is about building real skills—not just getting the right answers.