NYU Stern Graduate Summer Session #1
May 16th to July 3, 2019
Monday and Wednesday nights from 6-9pm

Programming in Python & Fundamentals of Software Development (INFO-GB 2335)

Learn how to code!

Today's businesses rely on application software to perform operations, aid decision-making, and drive competitive advantage. In this course, students will learn how to write practical business applications in the Python programming language. No prior programming experience is required. Students will also explore business models and best practices involved in the production and management of application software. Throughout the semester, students will be immersed in hands-on Python programming projects and should emerge with marketable technology skills.

For more information, contact Professor Rossetti at mrossett@stern.nyu.edu.

Programming in Python & Fundamentals of Software Development (INFO 2335)

University:	New York University
School:	Stern School of Business
Department:	Information, Operations, and Management Sciences (IOMS) / Info Systems (INFO)
Course:	Programming in Python & Fundamentals of Software Development (2335)
Section:	Mondays and Wednesdays from 6pm to 9pm in Kaufman Mgmt Center 460
Credits:	3
Prerequisites:	N/A

Description

Today's businesses rely on application software to perform operations, aid decision-making, and drive competitive advantage. In this course, students will learn how to write practical business applications in the Python programming language. No prior programming experience is required. Students will also explore business models and best practices involved in the production and management of application software. Throughout the semester, students will be immersed in hands-on Python programming projects and should emerge with marketable technology skills.

Learning Objectives

1. Learn how to write, debug, execute, and test command-line applications in the Python programming language.
2. Create application software to serve customers and automate business processes.
3. Identify and discuss revenue models and distribution models related to the production and consumption of application software.
4. Understand the business impacts of software licensing, open source software, and crowdsourcing.
5. Discuss security, privacy, and ethical considerations relevant in designing and managing computer-based information systems.
6. Follow software development best practices like version control and automated testing, and discuss their business implications.
7. Gain marketable programming skills and build an online portfolio of projects.
8. Have fun!

Community

Students

This course has a maximum enrollment of 50 students. Most are graduate business school (MBA) students, but enrollment is open to students from other NYU graduate schools and programs as well.

Professor

Adjunct Professor Michael Rossetti, a professional data scientist and software developer, will be teaching this course. Students should feel free to direct questions to the professor by sending a Slack direct message to @prof-rossetti or an email to mrossett@stern.nyu.edu. If emailing, all parties should use university-issued addresses. The professor aims to respond to messages within around one to three business days.

When sending announcements and replying to students, the professor may send messages outside of normal business hours. There is no expectation for students to keep the same schedule. Students should feel free to read and reply to messages at whatever time is most preferable for them!

Materials

Texts

Students are encouraged to leverage online resources and documentation such as:

- Python Documentation (Python.org)
- Python Tutorial (Python.org)
- Python Essential Training (Lynda.com)
- Git Documentation (Git-scm.com)
- Git and GitHub Learning Resources (GitHub.com)

Students may optionally reference technology books such as:

- Think Python (2e), by Allen B. Downey
- Pro Git, by Scott Chacon and Ben Straub

For additional context, students may optionally reference best-selling books such as:

- The Lean Startup, by Eric Ries
- Rework, by Jason Fried and David Heinemeier Hansson
- The Design of Everyday Things, by Don Norman

Computers

Each student is encouraged to configure their own personal computer with a Python development environment (see “Software” section below) at the beginning of the semester as instructed by the professor, and to bring this computer to each class. Mac and Windows operating systems should each provide a suitable development environment. However chromebooks, tablets, and netbooks may prove problematic and previous students have advised against them.

Students should NOT install Python or any other programs until instructed to do so by the professor.

Software

This course will introduce students to a standard set of software development tools, including: a development-class text editor like *VS Code*, *Atom*, *Sublime*, or *Notepad++*; and command-line utilities such as *Anaconda*, *Python*, *Pip*, and *Git*. Students should WAIT to install these programs until instructed by the professor, in the manner prescribed.

Students who wish to use their own preferred tools (like alternative text editors or IDEs) may do so as long as they are able to meet course expectations, but they should be aware the instructors may not be able to provide the same level of support for those tools.

Operations

NYU Classes

All enrolled students should have access to the course site in NYU Classes. The course calendar in NYU Classes is the most up-to-date source of information about the scheduling of class sessions and deliverables. The professor will also send announcements through NYU Classes and distribute all grades through the NYU Classes gradebook.

GitHub

GitHub is the leading online platform for sharing software and code-related resources. The course GitHub repository is the primary source for course materials, including programming language references, instructional exercises, and project descriptions.

Students will also use GitHub to submit project deliverables. When instructed by the professor, and in the manner prescribed, students should create a GitHub account as necessary and share their GitHub username with the professor to associate themselves with their work product. Deliverables will be submitted in public repositories by default. Student GitHub usernames and profiles need not contain any personally-identifiable information, but any student who desires additional privacy should consult with the professor within a week of enrolling, and the professor will recommend certain alternatives such as submission via private repositories.

Slack

Slack is an integrated chat platform that will help facilitate course communications. When invited by the professor, students should join the course Slack organization.

The professor will share code snippets and helpful links in the **#2335** channel on a regular basis, and students are encouraged to ask questions and help each other in that channel as well. The professor will share links to class recordings in the **#2335-media** channel, and may create additional channels as applicable to serve assignment-specific purposes or facilitate group communications.

Reference: Emoji Cheat Sheet 😃

Schedule

The schedule is tentative and may change to reflect actual pace of instruction. In the event of a major schedule change, the professor will likely send an announcement.

Module I: Python Programming and Applications

Day	**Date**	**Unit**	**Topic(s)**

Mon	May 20	1	Intro to Information Systems and Application Software;
			Command-line Computing;
			Python Development Environment Setup

Wed	May 22	2	Python Language Overview; Python Programming: processing data in-memory
Mon	May 27	N/A	Memorial Day (No Class)
Wed	May 29	3	Information Security, Privacy, and Ethics; Python Programming: processing user inputs
Mon	June 3	4	Datastores; Python Programming: processing data from CSV files
Wed	June 5	5	Networks, HTTP, and Application Programming Interfaces (APIs); Python Programming: processing data from the Internet

Module II: Management of Software Development

Day	**Date**	**Unit**	**Topic(s)**
Mon	June 10	5	Quality Control and Automated Tests; Python Programming: testing

Wed	June 12	7	Licensing, Intellectual Property, and Open Source;
			Version Control and Git;

Mon	June 17	8	Design Thinking Workshop;

			The Systems Development Lifecycle (SDLC); Survey of Third-party Python Packages
Wed	June 19	9	Project Development Support (Lab)

Mon	June 24	10	Delivery and Distribution;
			Hardware, Deployment Environments, and Server Management;
			Heroku

Recap and Review

Day	**Date**	**Unit**	**Topic(s)**
Mon	June 26	11	Retrospective Exercise; Final Exam Preparation Session
Wed	July 1	12	Final Exam Period

Evaluation

The NYU Classes Calendar is the authoritative source of information about due dates and weights of all items due for evaluation. However the table below provides a tentative outline.

Category	**Deliverable**	**Weight**	**Due**
Assignment	Onboarding Survey	3%	TBA
Assignment	Weekly Check-ins	2% * 6 = 12%	TBA
Project	“Shopping Cart” App	8%	TBA
Project	“Executive Dashboard” App	10%	TBA
Project	“Robo Advisor” App	10%	TBA
Assignment	“Freestyle” Project Proposal	4%	TBA
Project	“Freestyle” Project	18%	TBA
Exam	Final Exam	25%	TBA
Assignment	Industry Insights	10%	TBA

The professor aims to evaluate submissions and return grades within around seven days from the due date, and may utilize graduate assistants during the grading process. Any student who has a question or concern about a grade should ask the professor in writing within seven days of receiving the grade, and the professor will look into the matter in a timely manner.

Additionally, the professor reserves the right to award extra credit in recognition of valuable student participation and deliverables which exceed expectations.

Projects

Transaction Processing System

The Transaction Processing System (a.k.a. “Shopping Cart” Project) will facilitate a local corner grocery store’s customer checkout process. Students will write an interactive application which prompts an employee to scan grocery items, then calculates the total amount due and prints an itemized receipt. The software may interface with a real-life laser barcode scanner.

Management Information System

The Management Information System (a.k.a. “Executive Dashboard” Project) will provide an online retail business with reporting capabilities to aid decision-making. Students will write an application to transform monthly sales data into a summary report of business insights, including the aggregation of total sales and identification of top-selling products. The program may produce charts and graphs to help tell a compelling story.

Decision Support System

The Decision Support System (a.k.a. “Robo Advisor”) will provide an investment firm with the capability to automate its financial advisory processes. Students will create an application to generate stock market trading recommendations based on real-time historical stock market data from the Internet, and user inputs such as risk preferences. The system will issue HTTP requests and parse JSON-formatted responses.

Self-Directed Project

The Self-Directed (a.k.a. “Freestyle Advisor”) Project provides students with the flexibility to follow their own interests by proposing and ultimately implementing their own application software. First students will brainstorm and submit a proposal outlining their project’s scope, objectives, and requirements. Then students will implement the requirements by writing their own Python program. The final project deliverable will include not only the software itself, but also accompanying version history, documentation, and automated tests.

Proposal Phase

During the project proposal phase, students will define project scope and objectives and submit this information to the professor for approval. After reviewing the proposals, the professor may offer suggestions to help refine project focus, share helpful resources, and/or provide other guidance to help students succeed.

Implementation Phase

During the project implementation phase, students will write application software in Python. The software should transform information inputs into information outputs to achieve stated objectives as outlined in the project proposal. The software should strive to demonstrate a unique set of functionality which differentiates it in some significant way from other potential student submissions. If building upon an example project, the software should strive to differentiate itself from the example and/or add upon the example in a significant way.

Final Exam

The Final Exam is designed to evaluate student knowledge of Python programming concepts, software best practices, and technology management concepts. The exam will be administered during a 90-minute portion of the final class period. The exam will be administered in paper format, so students should remember to bring a pen or pencil. Details about the contents of the final exam will be announced during a final exam preparation session.

Policies

Attendance

All students are encouraged to attend class in-person. If not able to attend class in-person, students are encouraged to participate remotely via Slack, if possible.

Any student who is absent from class may risk missing in-class activities and deliverables. There will be no opportunities to make-up in-class assignments or exams due to absence, except with permission from the professor. To obtain permission for planned absences, a student must email the professor within the first week of class. The professor may grant make-up opportunities for unplanned absences only in the event of significant life events, in which case the professor reserves the right to request further documentation.

Instructional Continuity

If for any reason a class session is not able to be held in-person, the professor will implement a customized instructional continuity plan, which may include remote instruction via Slack.

Late Submissions and Extensions

Late submissions are generally not accepted. However, students may request a due date extension in response to extraordinary circumstances. Any student who requests a due date extension should email the professor days in advance of the original due date. Students should expect to submit deliverables on time unless the professor explicitly approves their extension request in writing, in which case late penalties may apply.

Learning Accommodations

Any student requiring learning accommodations, such as longer exam periods, should register and coordinate through the Moses Center within a week of enrolling.

Code of Conduct

All members of the learning community should at all times abide by the university’s Code of Ethical Conduct and the school's Code of Conduct.

Academic Integrity

Students are expected to follow the university's Academic Integrity Policy as well as those set forth here.

Although students are encouraged to work with each other to discuss and solve problems, submission of identical or nearly identical work may be seen as an academic integrity infraction. And although students are encouraged to leverage Internet resources, submission of work product generated by any other person may also constitute an infraction. Furthermore, if one student violates academic integrity policies by submitting the work product of another student, both students may be considered in violation and subject to penalties.

As a rule of thumb, it is each student's responsibility to type and understand every line of code submitted for evaluation. In situations where lines of boilerplate code or shared code are included in a submission, it is the responsibility of the student to accompany such code with one or more lines of "comments" which include a source link or other manner of attribution (e.g. "adapted from source: https://stackoverflow.com/a/36156/670433" or "jjones@stern.nyu.edu suggested this looping strategy"). However, students should know that submissions comprised of significant portions of code obtained in this way, even if properly attributed, may still constitute an infraction.

Any questions about what constitutes an academic integrity infraction should be proactively directed to the professor; retroactive naivete is not acceptable. Violations of academic integrity will be forwarded to the Academic Integrity Board, and may lead to consequences such as failure or dismissal.

Acknowledgement and Authorization

Class sessions will be recorded using university-administered systems and equipment. Links to these videos will be shared with students as soon as they become available, and may also be posted to the course repository on GitHub.

Students should be aware that class recordings may include their image, name, and voice. Any student who would like to opt out of class recordings should email the professor within a week of enrolling, and the professor will suggest some reasonable accommodations, which may include sitting in designated areas.