Programming in Python & Fundamentals of Software Development

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. Write, debug, and execute command-line applications in the Python programming language.
2. Create application software to perform 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!

Materials

Texts

Students may consult texts such as:

- The Lean Startup, by Eric Ries
- Rework, by Jason Fried and David Heinemeier Hansson
- Think Python (2e), by Allen B. Downey
- Pro Git, by Scott Chacon and Ben Straub

Students are also encouraged to reference online documentation and resources 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)

Computers

Each student should have access to a personal computer during class. Any student who doesn't own a personal computer may inquire about loaning a laptop from the library.

The computer should allow installation of Python, third-party packages and command-line utilities, and other software such as a text editor. Both Mac and Windows operating systems should provide a suitable development environment, but students with Windows computers may face additional and/or alternate instructions. Students should not install Python or any other programs until instructed to do so by the professor.

Schedule

Class No.	Topics / Usage	Due for Evaluation
1	Course Introduction; Information Systems, Software, and Python within a Business Context; Application Software Revenue and Distribution Models	Student Survey (5%)
2	Command-line Computing; Python Environment Setup; Software Version Control; Command-line Git	
3	Guided Tour of the Python Language I: Variables and Functions; Basic Datatypes (e.g. booleans, strings, and numbers); Advanced Datatypes (e.g. lists and dictionaries)	
4	Guided Tour of the Python Language II: Handling and Validating User Inputs; Handling Errors; Classes and Object-Oriented Programming (OOP); Reading and Writing Text Files and CSV Files	Management Information System (7.5%)
5	Systems Scope and Requirements;	

	Data Flow Diagramming; Process Diagramming	
6	Software Maintenance and Quality Control; Automated Testing; Test-driven Development; Code Simplification (Refactoring) and Review	Point-of-Sale System (12.5%)
7	Survey of Third-party Python Packages; Application Programming Interfaces (APIs); Internet (HTTP) Requests;	
8	Intellectual Property Models; Software Licensing; Open Source Software; Crowdsourcing	Inventory Management System (15%)
9	Deployment Environments; Server Management; Continuous Delivery	Self-Directed Project Proposal (5%)
10	Security, Privacy, and Ethics; Hacking; Malware	
11	Retrospective Exercise; Final Exam Preparation Session	Self-Directed Project Implementation (15%), Documentation (5%), and Tests (5%)
12	Final Exam Period	Final Exam (30%)

Evaluation

Student learning will be evaluated through hands-on Python programming projects and a final written Python exam.

Projects

Each of the first three projects asks students to write Python application software to modernize the technology capabilities of a fictitious example local corner grocery business. Whereas the final self-directed project challenges students to follow their own interests and imaginations.

1. Management Information System
2. Point-of-Sale (PoS) System
3. Inventory Management System
4. Self-Directed Project

Management Information System

The Management Information System will provide the business with reporting capabilities to aid decision-making. Students will write a script to transform monthly sales data into a summary report of business insights, including the aggregation of total sales and identification of top-selling products.

Point-of-Sale System

The Point-of-Sale System will enable the business' customer checkout process. Students will write an interactive command-line application which prompts an employee to scan grocery items, then calculates the total amount due and prints an itemized receipt.

Inventory Management System

The Inventory Management System will provide the business with the capability to manage its inventory of grocery products. Students will create an interactive command-line application which prompts an employee to list all existing products, search for a specific product, edit product information, or create new products. The system will read and write data in comma-separated values (CSV) format.

Self-Directed Project

The Self-Directed 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 documentation and automated tests.

Proposal Phase

During the project proposal phase, students will define project scope and objectives, including information inputs and outputs, 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.

Any student who has difficulty brainstorming a project idea will have the option to choose from a list of examples, including a stock-market purchase recommendation system. The stock-market purchase recommendation system allows a user to specify risk-tolerance preferences and stock symbols of interest, then processes historical stock price information from the Internet to provide the user with a purchase recommendation.

Implementation Phase

During the project implementation phase, students will write application software in Python. The software must 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 one of the example projects, 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.