

Advanced Programming Techniques with Python® (Second Edition) Bridge Document (From Advanced Programming Techniques with Python®)

This bridge document is written for instructors who have used Logical Operations' *Advanced Programming Techniques with Python®* courseware and are looking to come up to speed on the *Advanced Programming Techniques with Python® (Second Edition)* courseware quickly and efficiently. Our instructional designers work to retain sequencing and activities wherever possible, while adding new content to stay up to date on changes to the Python programming language, as well as provide an excellent class experience.

Overview of Changes

The *Advanced Programming Techniques with Python® (Second Edition)* course:

- Reflects the most recent version of the Python programming language as of development, Python 13.0, released in October 2024.
- Uses the most recent version of PyCharm as of development, PyCharm 2024.2.3, released in September 2024.
- Provides a Linux virtual machine (VM) as part of the data files for the activity environment. Students run PyCharm in this VM, and some activities have been modified to account for OS differences between Linux and Windows.
- Overhauls the code in content and activities to be cleaner, more consistent, and better formatted to meet Python style guidelines. This includes, but is not limited to:
 - Better use of whitespace and quotation marks.
 - More descriptive variables names.
 - Use of general or selective imports rather than universal imports.
- Reformats activity steps so that the step itself includes minimal instructions, and the space after each step includes the explanations.
- Restructures some activities to improve the overall flow, including splitting or merging certain steps.
- Reworks the structure/naming of data files, including moving snippets files to their own folder and changing the name of project folders/files to be more consistent with the Intro course.
- Condenses the amount of information presented on most slides, except for activity slides, which now summarize the activity scenarios.
- Incorporates the latest Logical Operations slide template, which has gone through a visual overhaul, and most notably, changed from standard 4:3 format to widescreen 16:9 format.
- Incorporates a total of one additional hour of material. This is an estimate and may vary depending on certain factors, like whether or not students copy code from the snippets files vs. typing the code themselves.

Lesson-Level and Topic-Level Structural Changes

The following table compares the lesson-level and topic-level outline of the *Advanced Programming Techniques with Python® (Second Edition)* course to the original *Advanced Programming Techniques with Python®* course. Most of the concepts contained in each lesson and topic are generally the same,

with a few additions and removals. The preponderance of changes is to the way the concepts and code are presented.

Change color key:

- Topics with minor updates
- Topics with significant updates

Advanced Programming Techniques with Python®		
Lesson	094022: First Edition	094032: Second Edition
1	Selecting an Object-Oriented Programming Approach for Python Applications <ul style="list-style-type: none"> A. Implement Object-Oriented Design B. Leverage the Benefits of Object-Oriented Programming 	Designing Object-Oriented Applications <ul style="list-style-type: none"> A. Select an Object-Oriented Programming Approach B. Leverage the Benefits of Object-Oriented Programming
2	Creating Object-Oriented Python Applications <ul style="list-style-type: none"> A. Create a Class B. Use Built-in Methods C. Implement the Factory Design Pattern 	Creating Object-Oriented Applications <ul style="list-style-type: none"> A. Create Classes B. Use Built-In Methods C. Implement the Factory Design Pattern
3	Creating a Desktop Application <ul style="list-style-type: none"> A. Design a Graphical User Interface (GUI) B. Create Interactive Applications 	Creating Desktop Applications <ul style="list-style-type: none"> A. Design Graphical User Interfaces (GUIs) B. Create Interactive Applications
4	Create Data-Driven Applications <ul style="list-style-type: none"> A. Connect to Data B. Store, Update, and Delete Data in a Database 	Creating Data-Driven Applications <ul style="list-style-type: none"> A. Connect to Databases B. Manage Data in Databases
5	Creating and Securing a Web Service-Connected App <ul style="list-style-type: none"> A. Select a Network Application Protocol B. Create a RESTful Web Service C. Create a Web Service Client D. Secure Connected Applications 	Creating and Securing Web-Based Applications <ul style="list-style-type: none"> A. Program Network Communications B. Create RESTful Web Services C. Create Clients for Web Services D. Secure Connected Applications
6	Programming Python for Data Science <ul style="list-style-type: none"> A. Clean Data with Python B. Visualize Data with Python C. Perform Linear Regression with Machine Learning 	Programming Python for Data Science <ul style="list-style-type: none"> A. Clean Data B. Visualize Data C. Implement Basic Machine Learning Using Regression
7	Implementing Unit Testing and Exception	Implementing Exception Handling and Unit

Advanced Programming Techniques with Python®		
Lesson	094022: First Edition	094032: Second Edition
	Handling <ul style="list-style-type: none"> A. Handle Exceptions B. Write a Unit Test C. Execute a Unit Test 	Testing <ul style="list-style-type: none"> A. Handle Exceptions B. Write Unit Tests C. Execute Unit Tests
8	Packaging an Application for Distribution <ul style="list-style-type: none"> A. Create and Install a Package B. Generate Alternative Distribution Files 	Packaging Applications for Distribution <ul style="list-style-type: none"> A. Create Python Packages B. Generate Executable Files

Content-Level Changes

The following structural changes were made at the content level (organized by the structure in the Second Edition):

- Lesson 1:
 - **Topic A:** Added/improved several code examples and diagrams primarily using a car analogy for object-oriented programming (OOP). Also changed guidelines to be more conceptual, in line with what is taught in the topic.
 - **Topic B:** Added several diagrams continuing the car analogy. Also rewrote/improved the discussion of inheritance, along with the `super()` function. Method *overriding* is now referred to consistently throughout the course (instead of overloading).
- Lesson 2:
 - **Topic A:** Much of the content has been rewritten so it's closer to a refresher of the Intro course (i.e., pared down but still covers much of the same material). Some new material has also been added, including discussions of dynamic class structure, public components, name mangling, and property usage. Some material was removed as being redundant or out of place, including discussions of destructor methods, modules, and f-strings. In the activity, the identifier `type` was changed to `form` to avoid shadowing issues, and the getters/setters are now properties with decorators.
 - **Topic B:** Added discussion of method overriding and added symbol equivalents for comparison methods. Removed binary arithmetic operators from table of numeric methods, as binary arithmetic is not taught in either course. In the activity, removed the part of the code where students add a destructor. Also, students now compare `Plywood` instances instead of `Wood` instances. The `wood_test.py` script is reused here and in the next topic.
 - **Topic C:** In the activity, added more explanation for why students are creating a factory and its benefits.
- Lesson 3:
 - **Topic A:** Reordered the content so frames and widgets are discussed first, then an example is given. Fleshed out the code more with widget examples and provided alternative approach to creating a frame. Also changed some of the design tools that are mentioned. In the second activity, provided a starter file for students that already has the data included, so they aren't required to type it out. Also renamed widget variables to be more descriptive.
 - **Topic B:** Reordered some of the content and cleaned up and rewrote quite a bit of code.
- Lesson 4:
 - **Overall:** Discussion and use of MySQL has largely been replaced by SQLite in both content and activities. SQLite is built in, lightweight, and simpler to set up/use for uncomplicated projects. MySQL is still mentioned as an alternative with its own pros and cons.
 - **Topic A:** Condensed and reworked discussion of file-based data sources and cloud services. Also added fetch methods to table. In the first activity, the SQLite database has already been created, so students don't need to run a script to create it. In the second activity, one database with two tables is used instead of two databases with one table each.

- **Topic B:** Removed discussion of Python data structures as not being relevant here. Added a discussion of table creation and renamed/reworded SQL injection material to focus on placeholders, with security being secondary. In the second activity, changed the query execution to use `fetchall()` and a `for` loop instead of `fetchone()` with a `while` loop. In the third activity, provided code with pre-written import statements, and used `today.isoformat()` as `today()` by itself is deprecated with SQLite.
- Lesson 5:
 - **Topic A:** Removed first figure as it didn't illustrate the main concept and introduced terms not taught.
 - **Topic B:** Expanded on example API and corrected discussions of SOAP and REST. Also added discussion of web frameworks and rewrote/reordered Django/Flask discussions. In the activity, provided most of the database code for students so they don't have to type it themselves.
 - **Topic C:** Included discussion of the `requests` module and simplified some examples. In the activity, the code students needed to delete has now been deleted for them.
 - **Topic D:** Simplified, rewrote, and rearranged much of the content. Removed the discussion of specific security tools as being too volatile to maintain, instead pointing to a GitHub repo that compiles a list of security tools for Python. In the activity, changed the client password to match the password of the VM, and changed the authentication verification code in `wdc_service.py`.
- Lesson 6:
 - **Overall:** Much of the content and activities in this lesson have been rewritten.
 - **Topic A:** Removed discussion of non-Python tools as not being relevant to this course. In the activity, `2020-Sales.csv` is now `orders.csv`; the `Cost` column is now `COGS`; and reworked the activity flow/steps and code.
 - **Topic B:** In the activity, the `subplots()` method is now used for plotting, and added more conclusions/explanations for what the plots show.
 - **Topic C:** In the activity, students now build a model using the same dataset they cleaned and visualized in the prior topics; the objective being to predict profits based on sales. Also removed step for changing the random seed in the holdout split, and added a new last step to visualize a single prediction on unseen data.
- Lesson 7:
 - **Topic A:** Reworked much of the content in this topic, and added new discussions of the call stack and how it relates to exception propagation. In the first activity, added more explanation of call stack and related errors. In the second activity, specified the exception class to handle in the code, and skipped steps of writing code to raise exceptions, since students just remove them later.
 - **Topic B:** Rearranged content for more logical order and added table of assert methods. Also reworked example of unit test. In the activity, added step to have students review the autogenerated test suite.
 - **Topic C:** Reworked discussion of test structure and removed guidelines that repeat prior topic's guidelines. Also moved the discussion of unit-test suites to Topic B where it's more appropriate and relevant to the activity. In the activity, students now modify the tests to make them more robust, focusing on testing the web service dynamically.

- Lesson 8:
 - **Topic A:** Rewrote discussion of dependencies and removed mention of `distutils` as it's been removed from Python. Also moved the first activity from Topic B to this topic, as it's more relevant here. In the first activity, included more files in the distribution. Also, students now add the dependent packages to the **setup.py** file.
 - **Topic B:** Changed topic to focus entirely on generating executables. Removed table of packaging options as they still require Python. Also, changed discussion of Windows executables to refer to executables more generically. Also removed the How To as the procedures are explained in the content. In the activity, removed the `-w` option from the `PyInstaller` command as it's not needed.
 - **Summary:** Changed second reflective question.

Additional Notes

- The Linux VM is provided in OVA format and meant to be used with Oracle VM VirtualBox, which is also included in the data files.
- All figures and screenshots have been updated.
- Non-screenshot images (and image elements) in the course PDF are now in vector format and retain full quality when scaled. This includes diagrams.
- The titling of some lessons, topics, knowledge blocks, and activities has changed to be more consistent and precise, even if the content in them has not changed.
- The assessment questions have changed.