

Introduction to Programming with Python® (Second Edition) Bridge Document (From Introduction to Programming with Python®)

This bridge document is written for instructors who have used Logical Operations' *Introduction to Programming with Python*® courseware and are looking to come up to speed on the *Introduction to Programming 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 *Introduction to Programming 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.
- Relies on more current and well-accepted Python conventions in content and activities, including:
 - Use of `with open...as` for opening/closing files (as opposed to using `open()` and `close()` explicitly).
 - Preference for using `pathlib` over `os.path` when dealing with file/directory paths.
 - Use of f-strings instead of `format()` for string interpolation.
 - More idiomatic usage of Boolean values in expressions (as opposed to `is True` or `is False`).
- 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 an additional 30 minutes of material per day, for a total of 1 additional hour.

Lesson-Level and Topic-Level Structural Changes

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

Change color key:

- Topics with minor updates
- **Topics with significant updates**

Introduction to Programming with Python®		
Lesson	094021: First Edition	094031: Second Edition
1	Setting Up Python and Developing a Simple Application <ul style="list-style-type: none"> A. Set Up the Development Environment B. Write Python Statements C. Create a Python Application D. Prevent Errors 	Setting Up Python and Developing a Simple Application <ul style="list-style-type: none"> A. Set Up the Development Environment B. Write Python Statements C. Create a Python Application D. Prevent Errors
2	Processing Simple Data Types <ul style="list-style-type: none"> A. Process Strings and Integers B. Process Decimals, Floats, and Mixed Number Types 	Preparing Simple Data Types <ul style="list-style-type: none"> A. Process Integers and Strings B. Process Decimals, Floats, and Mixed Number Types
3	Processing Data Structures <ul style="list-style-type: none"> A. Process Ordered Data Structures B. Process Unordered Data Structures 	Processing Data Structures <ul style="list-style-type: none"> A. Process Ordered Data Structures B. Process Unordered Data Structures
4	Writing Conditional Statements and Loops in Python <ul style="list-style-type: none"> A. Write a Conditional Statement B. Write a Loop 	Writing Conditional Statements and Loops <ul style="list-style-type: none"> A. Write Conditional Statements B. Write Loops
5	Structuring Code for Reuse <ul style="list-style-type: none"> A. Define and Call a Function B. Define and Instantiate a Class C. Import and Use a Module 	Structuring Code for Reuse <ul style="list-style-type: none"> A. Define and Call Functions B. Define and Instantiate Classes C. Import and Use Modules
6	Writing Code to Process File and Directories <ul style="list-style-type: none"> A. Write to a Text File B. Read from a Text File C. Get the Contents of a Directory D. Manage Files and Directories 	Writing Code to Process Files and Directories <ul style="list-style-type: none"> A. Write to Text Files B. Read from Text Files C. Get the Contents of Directories D. Manage Files and Directories
7	Dealing with Exceptions <ul style="list-style-type: none"> A. Handle Exceptions B. Raise Exceptions 	Handling Exceptions <ul style="list-style-type: none"> A. Handle Built-In Exceptions B. Handle Custom Exceptions

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:** Improved explanation of interpreters, compilers, and CPython. Both activities were altered to account for Linux operations vs. Windows operations.
 - **Topic B:** Updated keywords table and added floor division to arithmetic operators table. Removed discussion of functions. In activity, replaced using the `print()` function with retrieving string literals directly from console.
 - **Topic C:** Moved discussion of functions here as it is more appropriate when talking about Python applications than when introducing basic statements (like in prior topic).
- Lesson 2:
 - **Topic A:** The terms “indexing” and “slicing” are now consistently used throughout the course to refer to retrieving a single element from a data object (e.g., a string) and retrieving consecutive elements from a data object, respectively. Also added discussion of f-strings and removed discussion of formatting operator (outdated).
 - **Topic B:** Removed discussion of long integers (redundant).
- Lesson 3:
 - **Topic A:** Added results column for list-based tasks and added `not in` operator. Also added discussion of differences between `sort()` and `sorted()`, list copying, and guidelines for using ordered structures.
 - **Topic B:** Added guidelines for using unordered structures. In activity scenario, removed mention of using a set, as it’s not actually used in the activity (but appears in the next lesson).
 - **Summary:** Changed second reflective question.
- Lesson 4:
 - **Topic A:** Added discussion of Booleans and assignment operators, and changed explanation of identity operators to be more up to date. Also added discussion of the `match...case` statement.
 - **Topic B:** Added discussion of range, dictionary, and nested loops. Also added discussion of list comprehensions. In “Writing `for` Loops” activity, changed `for` loop in last block of code to be more concise.
- Lesson 5:
 - **Topic A:** Added discussion of parameters vs. arguments, passing arguments, and default values. In activity, renamed functions to align with finished program.
 - **Topic B:** Added real-world examples of class usage. Removed discussion of dynamic class structure (potentially confusing). Changed discussion of properties to focus on using decorators, as is now preferred. Also added diagrams for explaining inheritance and moved discussion of class scope earlier in topic.
 - **Topic C:** Added discussion of `import...as` and added example of renaming during a selective import. Expanded discussion of external modules/packages. Also removed `itertools` and `functools` from table of modules, while adding `pathlib` and `glob`. In activity, students now import `collections.Counter` and use it to perform the word-counting logic, rather than importing `os` and keeping the custom counting logic.

- Lesson 6:
 - **Overall:** Much of the code in this lesson has been refactored/rewritten to account for usage of `pathlib` and usage of `with open...as`.
 - **Topic A:** Switched order of discussing `write()` method and discussing paths and directories.
 - **Topic B:** Discussion of file-metadata timestamps now focuses on Linux conventions, with a note about Windows.
 - **Topics C and D:** Content/activities now focus on usage of `pathlib` and `shutil` rather than `os.path` and `shutil`.
 - **Summary:** Changed first reflective question.
- Lesson 7:
 - **Overall:** Reworded lesson/topic titles. Also wrote more specific exception handling rather than using bare `except` branches.
 - **Topic A:** Added discussion of exception hierarchy and rewrote discussion of built-in exception classes. Also added discussion of exception ordering and removed discussion of `finally` branches (more suitable in advanced course).
 - **Summary:** Changed both reflective questions.
- Appendix B:
 - **Topic A:** Added item about not surrounding `=` with spaces when used to assign arguments/parameters.

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.
- Most figures and all 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 knowledge blocks and activities has changed to be more consistent and precise, even if the content in them has not changed.
- Some longer knowledge blocks have been split into multiple, smaller knowledge blocks for more logical delivery.
- The assessment questions have changed.