Preface

This exercise book was born from the Python courses offered by the IT Education Center at Università Bocconi starting from 2017. Over time, many materials have been produced which have been collected, expanded, refined and organized here with the aim of providing an effective and progressive support for university programming courses starting from the basic level, and for anyone who, independently or as part of a structured course, approaches programming in Python with little or no previous experience.

The workbook allows you to practice the use of a large variety of constructs, objects and operators (including loops, conditional expressions, built-in and custom-created functions, classes, standard and third-party library modules, exceptions, objects such as strings, lists, tuples and dictionaries) that allow you to understand the basics of programming in Python (version 3.x) and to create simple programs by yourself.

We start from the basics in unit 1, dedicated to the basics of Python, to progressively arrive at dealing with structured and complex cases. The 12 thematic units contain from 5 to 10 questions and from 5 to 16 exercises each, for a total of 70 questions and 134 exercises, with the related solutions (correct answers to the questions and solved files for the exercises). Among these, 38 questions and 117 exercises are commented, in order to fully understand the solutions.

At the address http://mybook.egeaonline.it all the solved files are available (and the starting ones, if any) that allow you to carry out the exercises and verify the solutions.

To obtain the maximum effectiveness in the use of the workbook, it is advisable to combine it with a book that, in an equally gradual way, introduces the concepts of programming in Python (for instance Clerici A., De Pra M., Debernardi MC, Tosi D., *Learning Python*, Egea, 2020).

Please note the didactic intent of this volume: often, especially in the more complex exercises, there are several possible solutions, one of which is presented (the one deemed most relevant from a didactic point of view). There could be others, equally valid, and sometimes even more effective from the point of view of programming efficiency, but more difficult to understand for those starting from the basics.

Plan of the workbook

Unit 1. Python basics
Mathematical operators (+, -, *, /, **, %, //) and use on numbers and strings; variables, assignment and reassignment; escape codes (`\n`, `\t`, `\`); built-in functions (`help`, `print`, `format`, `input`, `type`, `import`, `abs`, `pow`, `max`, `min`, `round`), basic data types and conversion between data types (`int`, `float`, `str` functions).
(10 questions, 15 commented exercises)

Unit 2. Conditional statements
Conditional structures with `if`, `if-else`, `elif`; `pass` statement; Boolean values `True` and `False`; relational operators (==, !=, <, <=, >, >=); logical operators `and`, `or`, `not`; `in` and `not in` containment operators.
(6 commented questions, 11 commented exercises)

Unit 3. Loops
`for` loop, `while` loop, `range` function, `in` operator.
(5 questions, 15 exercises of which 6 commented)

Unit 4. Modules of the standard library
Import of the standard library modules; functions of `math` and `random` modules.
(7 questions, 10 exercises of which 6 commented)

Unit 5. Functions
Custom functions, local variables, docstring.
(5 questions, 10 exercises of which 6 commented)

Unit 6. Exceptions
Handling of generic exceptions with `try…except` and in specific cases with `NameError`, `TypeError`, `ValueError`, `ZeroDivisionError`.
(5 commented questions, 5 commented exercises)

Unit 7. Sequences: strings
Operations on strings, functions `len`, `str`, `max`, `min`, `sorted`; methods `find`, `count`, `upper`, `lower`, `capitalize`, `strip`, `replace`, `startswith`, `endswith`, `join`, indexing and slicing.
(5 commented questions, 15 commented exercises)

Unit 8. Sequences: lists and tuples
Lists and tuples; list methods (`append`, `clear`, `copy`, `count`, `extend`, `index`, `insert`, `pop`, `remove`, `reverse`, `sort`); tuple methods (`count`, `index`); `len`, `list`, `max`, `min`, `sorted`, `sum` and `tuple` functions; `random.choice` function; `in`, `not in`, `+` and `*` operators; traversing lists and tuples; two-dimensional lists (or lists of lists).
(5 commented questions, 16 commented exercises)

Unit 9. File access
File access, reading data from a file, writing data in a file; `open` and `close` functions; file access mode (read, write, append); `read`, `readline`, `readlines`, `writable`, `write` and `writelines` methods.
(5 commented questions, 5 commented exercises)

Unit 10. Dictionaries
Creation of dictionaries; adding and modifying elements; traversing; `in` and `not in` operators on keys; built-in functions `dict`, `len`, `sorted`; specific methods `keys`, `values`, and `items`; indexing by key and `get` method; `pop` and `popitem` methods and `del` operator; union of dictionaries and `update` method; complex structures nested within dictionaries, their access and traversing.
(6 commented questions, 11 commented exercises)

Unit 11. Classes
Creating classes and instances; `self` parameter; special methods `__init__` (constructor) and `__str__`; definition of instance and class attributes; creation of ad hoc methods; definition of class and method docstrings; use of the `None` value; calling methods within other methods; inheritance and `object` class; use of the constructor method of the parent/superclass in the child/subclass; calling of methods of the parent/superclass in the child/subclass; make attributes and methods invisible to the user.
(6 commented questions, 11 commented exercises)

Unit 12. Other modules
Standard library modules: `os`, `webbrowser`, `turtle`; third party modules: `pyperclip`, `openpyxl`, `matplotlib.pyplot` and related functionalities.
(5 questions, 10 commented exercises)