

Contents

List of figures — XIII

List of tables — XVII

1	Introduction to Python — 1
1.1	Why Python? — 1
1.1.1	Drawbacks of Python — 2
1.1.2	What is new in Python 3.11.0? — 2
1.2	History of Python — 4
1.3	Major features of Python — 4
1.3.1	What is new in the upcoming Python 3.11.0? — 5
1.4	Market demand — 10
1.5	Why Python in mobile app development? — 10
1.6	Python versions — 15
1.7	Architecture of Python application — 16
1.8	System requirement — 17
1.9	Installation of Python in various platforms — 18
1.9.1	Installation in Windows — 18
1.9.2	Installation in Linux — 18
1.9.3	Installation in Mac — 19
1.10	Creating First Hello World App — 19
1.10.1	Basic operations in Python prompt — 23
1.11	Anatomy of Python program — 24
1.12	IDLE — 24
1.13	User input — 24
1.14	Installing external modules using pip — 25
1.15	What is PEP 8 — 26
1.16	PEP-8 formatting tools — 27
	Summary — 27
	Key terms — 28
	Review questions — 28
	Exercise — 29
2	Python basics — 31
2.1	Variables — 31
2.2	Data types — 33
2.2.1	Booleans — 35
2.2.2	Integer — 35
2.2.3	Float — 35
2.2.4	String — 36

2.2.5	None — 36
2.3	Block indentation — 36
2.4	Math calculations — 37
2.5	Operators — 38
2.6	String operations — 38
2.7	Indexing — 39
2.8	Slicing — 40
2.9	Concatenation — 40
2.10	Single-line and Multiline comments — 41
2.11	Functions — 41
2.11.1	Built-in functions — 42
2.11.2	User-defined functions — 43
2.11.3	Function with argument — 43
2.11.4	Flow of execution — 43
2.11.5	Default values — 44
2.11.6	*args and **kwargs in function — 44
2.12	Return value — 45
2.13	Type-casting — 46
2.14	Disassembler — 46
2.15	Lambda keyword — 47
	Summary — 48
	Key terms — 48
	Review questions — 49
	Exercise — 49
3	Conditions and loops — 52
3.1	If and else and conditional statements — 52
3.1.1	Nested if-else — 54
3.2	Loops — 55
3.2.1	For loop — 55
3.2.2	While loop — 56
3.3	Break and continue statement — 56
3.4	Collections — 58
3.4.1	Tuples — 58
3.4.2	Lists — 59
3.4.3	Sets — 61
3.4.4	Frozen sets — 63
3.4.5	Byte — 63
3.4.6	Byte array — 64
3.4.7	String — 64
3.4.8	Dictionaries — 66
3.4.9	Arrays — 68

- 3.5 Date and time — 69
 - Summary — 69
 - Key terms — 70
 - Review questions — 70
 - Exercise — 70

- 4 Object-oriented programming (OOP) — 73**
 - 4.1 Object-oriented programming (OOP) — 73
 - 4.2 Class — 73
 - 4.3 Object — 74
 - 4.4 Functions — 74
 - 4.5 Abstraction — 76
 - 4.5.1 Abstract class — 77
 - 4.5.2 Interface — 78
 - 4.6 Polymorphism — 81
 - 4.6.1 Overloading — 83
 - 4.6.2 Overriding — 85
 - 4.7 Encapsulation — 86
 - 4.8 Inheritance — 87
 - 4.9 Access modifier/Specifier — 88
 - 4.10 Decorator — 89
 - 4.10.1 @classmethod decorator — 90
 - 4.10.2 @staticmethod decorator — 91
 - Summary — 91
 - Key terms — 92
 - Review questions — 92
 - Exercise — 93

- 5 Standard libraries — 95**
 - 5.1 File operations using native — 95
 - 5.1.1 Writing file — 96
 - 5.2 OS Module — 96
 - 5.3 Exception handling — 98
 - 5.3.1 User-defined exceptions — 99
 - 5.4 Import standard Python modules and packages — 99
 - 5.4.1 Creating a module — 100
 - 5.4.2 Packages — 100
 - 5.5 Regular expression — 102
 - 5.6 Math module — 104
 - 5.6.1 Writing documentation for module — 105
 - 5.7 JSON — 106
 - 5.8 SQLite3 — 108

5.9	SQLAlchemy — 110
	Summary — 111
	Key terms — 112
	Review questions — 112
	Exercise — 113
6	Getting started with Kivy — 116
6.1	Why cross-platform? — 116
6.2	What is Kivy? — 116
6.3	Why Python in mobile app? — 116
6.4	Kivy architecture — 117
6.5	Python developed mobile apps — 120
6.6	History of Kivy — 121
6.7	Why not Cordova? — 121
6.8	Kivy versions — 121
6.9	How is it different from the cross-platform app framework? — 122
6.10	System requirement — 122
6.11	Installation of Kivy — 123
6.11.1	Installation in Windows — 123
6.11.2	Installation in Mac OS — 123
6.11.3	Installation in Linux — 123
6.11.4	What are Wheels and pip? — 126
6.11.5	Cython installation in Linux — 126
6.11.6	Verifying installation — 126
6.12	Creating the first Hello World app — 127
6.12.1	Event handling in Kivy — 127
6.12.2	Event dispatcher — 128
6.12.3	Main loop — 129
6.12.4	Custom events — 129
6.13	Anatomy of Kivy — 129
6.13.1	Running Kivy application — 132
6.13.2	Listing and uninstalling Kivy — 133
6.14	What is deprecated? — 134
6.15	What is new in widgets? — 134
	Summary — 134
	Key terms — 135
	Review questions — 135
	Exercise — 136
7	Kivy basics — 138
7.1	Configuring environment — 138
7.2	Configure Kivy and how to import Kivy library — 147

- 7.3 First app — 149
 - 7.3.1 Widgets — 153
 - 7.3.2 Custom widgets — 153
- 7.4 Orientation horizontal and vertical — 155
 - 7.4.1 size-hint and pos_hint — 158
 - 7.4.2 pos_hint — 160
 - 7.4.3 FileChooser — 163
- 7.5 super, padding, __init__, and add_widget — 166
 - 7.5.1 Super — 166
 - 7.5.2 __init__ — 168
 - 7.5.3 add_widget — 169
- Summary — 171
- Key terms — 171
- Review questions — 172
- Exercise — 172
- 8 Layouts — 175**
 - 8.1 What is the layout? — 175
 - 8.2 Box layout — 175
 - 8.3 Float layout — 179
 - 8.4 Anchor layout — 185
 - 8.5 Grid layout — 188
 - 8.6 Relative layout — 191
 - 8.7 Stack layout — 193
 - 8.8 Page layout — 194
- Summary — 196
- Key terms — 197
- Review questions — 197
- Exercise — 197
- 9 Designing user interfaces — 200**
 - 9.1 What is UI (user interface) components — 200
 - 9.2 Buttons — 200
 - 9.3 Labels — 203
 - 9.4 ListView — 206
 - 9.4.1 RecyclerView — 208
 - 9.5 Image — 212
 - 9.6 Events and properties — 214
 - 9.7 Fonts and their properties — 216
 - 9.8 Popup menu — 217
 - 9.9 TextInput — 219
 - 9.10 ScrollView — 220

9.11	Kivy clock — 221
	Summary — 222
	Key terms — 223
	Review questions — 223
	Exercise — 223
10	UX widgets — 226
10.1	ActionBar and slider — 226
10.2	CheckBox and text on Window — 228
10.2.1	Toggle button — 233
10.2.2	TreeView — 235
10.3	Radio/checkbox button and label management — 238
10.4	ProgressBar and carousel — 242
10.4.1	Kivy clock — 243
10.4.2	Custom ProgressBar — 245
10.4.3	Carousel layout — 247
10.4.4	Scatter — 251
10.5	Canvas — 253
10.6	AsyncImage — 256
10.7	Spinner — 257
10.8	Accordion — 258
10.9	Switching between two screens — 260
	Summary — 262
	Key terms — 263
	Review questions — 264
	Exercise — 265
11	Graphics handling — 267
11.1	Graphics — 267
11.2	Interacting with another framework — 270
11.3	Modules — 276
11.4	Network support — 277
11.5	Storage tool — 277
11.6	Rotate clock and rotate speed — 280
11.7	Triggered events — 281
11.8	Animations — 284
11.9	Audio — 285
11.10	Video — 287
11.11	Annotation — 292
	Summary — 292

Key terms —	294
Review questions —	295
Exercise —	295

12	Packaging app for various platforms —	298
12.1	Adding behavior with mixin class —	298
12.1.1	Packaging application —	300
12.1.2	Testing application with Android device —	301
12.1.3	Packaging for Android —	307
12.1.4	Plyer —	321
12.2	Building an Android APK on OS X —	322
12.3	Package licensing —	322
12.4	Packaging application using Kivy launcher —	323
12.5	External libraries —	323
12.6	Kivy application testing with PyUnit library —	323
12.7	Release on the market —	328
12.8	Kivy application testing with nose library —	328
	Summary —	329
	Key terms —	330
	Review questions —	331
	Exercise —	332

13	Version control using git —	335
13.1	Introduction of git —	335
13.2	Implementation of git —	335
13.2.1	Installation of git —	335
13.3	Sample projects —	340
13.3.1	Sample project 1 —	340
13.3.2	Sample project 2 —	344
13.4	Kivy catalog —	347
13.5	Project 1: Creating my own screen —	355
13.6	Project 2: game —	357
	Summary —	360
	Key terms —	361
	Review questions —	361
	Exercise —	362

14	Design patterns —	365
14.1	Introduction of design pattern —	365
14.1.1	Why design pattern —	366
14.2	Creational design pattern —	367
14.2.1	Singleton design pattern —	367

- 14.2.2 Prototype design pattern — **368**
- 14.2.3 Factory design pattern — **370**
- 14.2.4 Abstract factory method — **372**
- 14.2.5 Builder design pattern — **373**
- 14.3 Behavioral design pattern — **374**
- 14.3.1 Chain of responsibility design pattern — **374**
- 14.3.2 Iterator design pattern — **377**
- 14.3.3 Mediator design pattern — **378**
- 14.3.4 Memento design pattern — **380**
- 14.3.5 Observer design pattern — **381**
- 14.3.6 State design pattern — **382**
- 14.3.7 Strategy design pattern — **384**
- 14.3.8 Template design pattern — **385**
- 14.3.9 Visitor design pattern — **387**
- 14.3.10 Interpreter design pattern — **389**
- 14.4 Structural design pattern — **392**
- 14.4.1 Bridge design pattern — **392**
- 14.4.2 Adapter design pattern — **393**
- 14.4.3 Model view controller (MVC) design pattern — **395**
- 14.4.4 Composite design pattern — **398**
- 14.4.5 Decorator design pattern — **400**
- 14.4.6 Facade design pattern — **401**
- 14.4.7 Proxy design pattern — **402**
- 14.4.8 Object-oriented pattern — **404**
- 14.4.9 Command design pattern — **404**
- 14.4.10 Fly weight design pattern — **408**
- Summary — **409**
- Key terms — **410**
- Review questions — **411**
- Exercise — **411**

Bibliography — 415

Index — 417