PCPP1™ – Certified Professional in Python Programming 1

(Exam PCPP-32-101)
Last revised: (March 11, 2022)

Section 1: Advanced Object-Oriented Programming

Objectives covered by the section → 15 exam items

PCPP-32-101 1.1 – Understand and explain the basic terms and programming concepts used in the OOP paradigm

- essential terminology: class, instance, object, attribute, method, type, instance and class variables, superclasses and subclasses
- reflection: `isinstance()`, `issubclass()`
- the `__init__()` method
- creating classes, methods, and class and instance variables; calling methods; accessing class and instance variables

PCPP-32-101 1.2 – Perform Python core syntax operations

- Python core syntax expressions – magic methods: comparison (e.g., `__eq__(self, other)`), numeric (e.g., `__abs__(self)`), type conversion (e.g., `__int__(self)`), intro/retrospection (e.g., `__str__(self)`, `__instancecheck__(self, object)`), attribute access (e.g., `__getattr__(self, attribute)`), container access (e.g., `__getitem__(self, key)`)
- operating with special methods
- extending class implementations to support additional core syntax operations

PCPP-32-101 1.3 – Understand and use inheritance, polymorphism, and composition

- class hierarchies; single vs. multiple inheritance; Method Resolution Order (MRO)
- duck typing
- inheritance vs. composition; modeling real-life problems using "is a" and "has a" relations

PCPP-32-101 1.4 – Understand extended function argument syntax and use decorators

- `*args`, `**kwargs`; forwarding arguments; parameter handling
- closures; function and class decorators; decorating functions with classes
- creating decorators: decorator patterns, decorator arguments, wrappers; decorator stacking; syntactic sugar
- special methods: `__call__`, `__init__`

PCPP-32-101 1.5 – Design, build, and use static and class methods

- implementing class and static methods; `@classmethod` vs. `@staticmethod`; the `cls` parameter
- class methods: accessing/modifying class state, creating objects

PCPP-32-101 1.6 – Understand and use abstract classes and methods

- defining and implementing abstract classes/methods; overriding abstract methods
- multiple inheritance from abstract classes; delivering multiple child classes

PCPP-32-101 1.7 – Understand attribute encapsulation

- definition, meaning, usage; operating with getter, setter, deleter methods

PCPP-32-101 1.8 – Apply subclassing of built-in classes

- inheriting from built-ins to extend/modify features, methods, attributes

PCPP-32-101 1.9 – Advanced techniques for creating and serving exceptions

- exceptions as objects; named attributes; chained exceptions (`__context__`, `__cause__`); implicit vs. explicit chaining
- analyzing traceback objects (`__traceback__`)
- operating with different kinds of exceptions

PCPP-32-101 1.10 – Perform shallow and deep copy operations

- object: label vs. identity vs. value; `id()` and the `is` operator
- using `copy()` and `deepcopy()`

PCPP-32-101 1.11 – (De)serialize Python objects

- persistence, serialization, deserialization – meaning, purpose, usage
- single-byte-stream serialization: the `pickle` module; `dumps()`, `loads()`
- serialization dictionary via `shelve`: file modes, creating shelve objects

PCPP-32-101 1.12 – Explain the concept of metaprogramming

- metaclasses: meaning, purpose, usage; the `type` metaclass and `type()`
- special attributes: `__name__`, `__class__`, `__bases__`, `__dict__`; operating with metaclasses, class variables, and class methods

Section 2: Coding Conventions, Best Practices, and Standardization

Objectives covered by the section → 7 exam items

PCPP-32-101 2.1 – Python Enhancement Proposals and Python philosophy

- PEP concept and selected PEPs: PEP 1, PEP 8, PEP 20, PEP 257
- PEP 1 types, formats, purpose, guidelines
- PEP 20 aphorisms (`import this`) and guiding principles

PCPP-32-101 2.2 – Employ PEP 8 guidelines, conventions, and best practices

- PEP 8 checkers; code layout (indentation, continuation lines, max length, line breaks, blank lines)
- default encodings; module imports
- string quotes, whitespace, trailing commas
- comments vs. documentation strings
- naming conventions; programming recommendations

PCPP-32-101 2.3 – Employ PEP 257 conventions and related standards

- docstrings: rationale, usage; comments vs. docstrings
- PEP 484 type hints
- creating, using, and accessing docstrings (one-line vs. multi-line); documentation standards, linters, fixers

Section 3: GUI Programming

Objectives covered by the section → 8 exam items

PCPP-32-101 3.1 – Basic concepts and terminology for GUI programming

- GUI meaning/rationale; visual programming; widgets/controls (windows, buttons, icons, labels, etc.)
- classical vs. event-driven programming; events – basic terms; GUI toolkits

PCPP-32-101 3.2 – Use GUI toolkits, blocks, and conventions to build simple GUIs

- importing `tkinter`; `Tk()`, `mainloop()`, `title`
- adding widgets: buttons, labels, frames; `place()`; coordinates and sizing
- event controller: handlers, callbacks, `destroy()`, dialog boxes
- validating input and handling errors; `Canvas` methods
- `Entry`, `Radiobutton`, `Button`; layout with `grid` and `place`; binding events via `bind()`

PCPP-32-101 3.3 – Widgets and event handling

- geometry managers; coloring widgets (RGB, HEX)
- event-driven programming with events/callbacks
- widget properties/methods; observable variables and observers
- using clickable/non-clickable widgets; identifying/servicing events

Section 4: Network Programming

Objectives covered by the section → 8 exam items

PCPP-32-101 4.1 – Basic concepts of network programming

- REST; sockets; domains, addresses, ports, protocols, services
- connection-oriented vs. connectionless; clients and servers

PCPP-32-101 4.2 – Work with sockets in Python

- the `socket` module: creating sockets; connecting to HTTP servers; closing connections
- sending requests (`send()`); receiving responses (`recv()`); exception handling

PCPP-32-101 4.3 – Data transfer mechanisms

- JSON syntax/structure/types; `json` module (`dumps()`, `loads`) for Python↔JSON
- XML syntax/structure; DTD; XML as a tree; processing XML files

PCPP-32-101 4.4 – Design, develop, and improve a simple REST client

- the `requests` module; test environments
- HTTP methods: `GET`, `POST`, `PUT`, `DELETE`; CRUD operations
- adding/updating/fetching/removing data; analyzing responses; status codes

Section 5: File Processing and Communicating with a Program's Environment

Objectives covered by the section → 7 exam items

PCPP-32-101 5.1 – Database programming in Python

- `sqlite3` module; `connect`/`close` database connections
- creating tables; inserting, reading, updating, deleting data
- transaction demarcation; cursor methods: `execute`, `executemany`, `fetchone`, `fetchall`
- basic SQL statements (`SELECT`, `INSERT INTO`, `UPDATE`, `DELETE`, etc.)

PCPP-32-101 5.2 – Process different file formats in Python

- parsing XML: `find`/`findall`; building XML via `Element` and `SubElement`
- reading/writing CSV via `reader`, `writer`, `DictReader`, `DictWriter`
- logging events; logging levels; `LogRecord` attributes; custom handlers/formatters
- configuration files with `ConfigParser`; value interpolation in `.ini` files