PCEP™ – Certified Entry-Level Python Programmer (Exam PCEP-30-01) – EXAM SYLLABUS

PCEP-30-01 Exam

Status: Live & Retiring **(Dec 31, 2022)**

The exam consists of five sections:

Section 1 → 5 items	Max Raw Score: 17 (17%)
Section 2 → 6 items	Max Raw Score: 20 (20%)
Section 3 → 6 items	Max Raw Score: 20 (20%)
Section 4 → 7 items	Max Raw Score: 23 (23%)
Section 5 → 6 items	Max Raw Score: 20 (20%)

Last updated: May 25, 2021
Aligned with Exam PCEP-30-01

Section 1: Basic Concepts (17%)

Objectives covered by the block (5 exam items)

PCEP-30-01 1.1 – Understand fundamental terms and definitions

- interpreting and the interpreter, compilation and the compiler
- lexis, syntax, and semantics

PCEP-30-01 1.2 – Understand Python's logic and structure

- keywords
- instructions
- indentation
- comments

PCEP-30-01 1.3 – Use and understand different types of literals and numeral systems

- Boolean, integers, floating-point numbers
- scientific notation
- strings
- binary, octal, decimal, and hexadecimal numeral systems
- variables
- naming conventions

PCEP-30-01 1.4 – Choose operators and data types adequate to the problem

- numeric operators: ** * / % // + –
- string operators: * +
- assignment and shortcut operators

Section 2: Data Types, Evaluations, and Basic I/O Operations (20%)

Objectives covered by the block (6 exam items)

PCEP-30-01 2.1 – Build complex expressions and determine data type

- unary and binary operators
- priorities and binding
- bitwise operators: ~ & ^ | << >>

- Boolean operators: *not*, *and*, *or*
- Boolean expressions
- relational operators (== != > >= < <=)
- the accuracy of floating-point numbers

PCEP-30-01 2.2 – Perform complex Input/Output operations

- the *print()* and *input()* functions
- the *sep=* and *end=* keyword parameters
- the *int()*, *float()*, *str()*, and *len()* functions
- type casting

PCEP-30-01 2.3 – Operate on strings

- constructing, assigning, indexing, and slicing strings
- immutability
- quotes and apostrophes inside strings
- escaping using the \ character
- basic string functions and methods

Section 3: Control Flow – conditional blocks and loops (20%)

Objectives covered by the block (6 exam items)

PCEP-30-01 3.1 – Make decisions and branch the flow with the *if* instruction

- conditional statements: if, if-else, if-elif, if-elif-else
- multiple conditional statements
- nesting conditional statements

PCEP-30-01 3.2 – Perform different types of loops

- the *pass* instruction
- building loops with *while*, *for*, *range()*, and *in*
- iterating through sequences
- expanding loops with *while-else* and *for-else*
- nesting loops and conditional statements
- controlling loop execution with *break* and *continue*

Section 4: Data Collections – Lists, Tuples, and Dictionaries (23%)

Objectives covered by the block (7 exam items)

PCEP-30-01 4.1 – Collect and process data using lists

- constructing vectors
- indexing and slicing
- the *len()* function
- list methods: *append()*, *insert()*, *index()*, etc.
- functions: *len()*, *sorted()*
- the *del* instruction
- iterating through lists with the *for* loop
- initializing loops
- the *in* and *not in* operators
- list comprehensions
- copying and cloning
- lists in lists: matrices and cubes

PCEP-30-01 4.2 – Collect and process data using tuples

- tuples: indexing, slicing, building, immutability
- tuples vs. lists: similarities and differences
- lists inside tuples and tuples inside lists

PCEP-30-02 4.3 Collect and process data using dictionaries

- dictionaries: building, indexing, adding and removing keys
- iterating through dictionaries and their keys and values
- checking the existence of keys
- methods: *keys()*, *items()*, and *values()*

Section 5: Functions (20%)

Objectives covered by the block (6 exam items)

PCEP-30-01 5.1 – Decompose the code using functions

- defining and invoking user-defined functions and generators
- the *return* keyword, returning results
- the *None* keyword
- recursion

PCEP-30-01 5.2 – Organize interaction between the function and its environment

- parameters vs. arguments
- positional, keyword, and mixed argument passing
- default parameter values
- name scopes, name hiding (shadowing), and the *global* keyword