PCEP – Certified Entry-Level Python Programmer Certification: Exam Syllabus

PCEP-30-01 (Retiring December 31, 2022)

Exam block #1: Basic Concepts (17%)

Objectives covered by the block (5 exam items)

- fundamental concepts: interpreting and the interpreter, compilation and the compiler, language elements, lexis, syntax and semantics, Python keywords, instructions, indenting
- literals: Boolean, integer, floating-point numbers, scientific notation, strings
- comments
- the print() function
- the input() function
- numeral systems (binary, octal, decimal, hexadecimal)
- numeric operators: **** * / % // + –**
- string operators: *** +**
- assignments and shortcut operators

PCEP-30-02 (Active)

Exam block #1: Computer Programming and Python Fundamentals (18%)

Objectives covered by the block (7 exam items)

- **PCEP 1.1 Understand fundamental terms and definitions**
  interpreting and the interpreter, compilation and the compiler, lexis, syntax and semantics
- **PCEP 1.2 Understand Python's logic and structure**
  keywords, instructions, indenting, comments
- **PCEP 1.3 Introduce literals and variables into code and use different numeral systems**
  Boolean, integers, floating-point numbers, scientific notation, strings, binary, octal, decimal, and hexadecimal numeral system, variables, naming conventions, implementing PEP-8 recommendations
- **PCEP 1.4 Choose operators and data types adequate to the problem**
  numeric operators: ** * / % // + –, string operators: * +, assignments and shortcut operators, operators: unary and binary, priorities and binding, bitwise operators: ~ & ^ | << >>, Boolean operators: not and or, Boolean expressions, relational operators (== != > >= < <=), the accuracy of floating-point numbers, type casting
- **PCEP 1.5 Perform Input/Output console operations**
  print(), input() functions, sep= and end= keyword parameters, int() and float() functions

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

Objectives covered by the block (6 exam items)

- operators: unary and binary, priorities and binding
- bitwise operators: **~ & ^ | << >>**
- Boolean operators: **not and or**
- Boolean expressions
- relational operators (**== != > >= < <=**), building complex Boolean expressions
- accuracy of floating-point numbers
- basic input and output operations using the **input()**, **print()**, **int()**, **float()**, **str(), len()** functions
- formatting **print()** output with **end=** and **sep=** arguments
- type casting
- basic calculations
- simple strings: constructing, assigning, indexing, immutability

Exam block #2: Control Flow – Conditional Blocks and Loops (29%)

Objectives covered by the block (8 exam items)

- **PCEP 2.1 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 2.2 Perform different types of iterations**
  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

Exam block #3: Control Flow – loops and conditional blocks (20%)

Objectives covered by the block (6 exam items)

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

Exam block #3: Data Collections – Tuples, Dictionaries, Lists, and Strings (25%)

Objectives covered by the block (7 exam items)

- **PCEP 3.1 Collect and process data using lists**
  constructing vectors, indexing and slicing, the len() function, basic list methods (append(), insert(), index()) and functions (len(), sorted(), etc.), the del instruction; iterating through lists with the for loop, initializing loops; in and not in operators, list comprehensions; copying and cloning, lists in lists: matrices and cubes
- **PCEP 3.2 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 3.3 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; keys(), items() and values() methods
- **PCEP 3.4 Operate with strings**
  constructing strings, indexing, slicing, immutability; escaping using the \ character; quotes and apostrophes inside strings, multi-line strings, basic string functions and methods

Exam block #4: Data Collections – Lists, Tuples, and Dictionaries (23%)

Objectives covered by the block (7 exam items)

- simple lists: constructing vectors, indexing and slicing, the **len()** function
- lists in detail: indexing, slicing, basic methods (**append()**, **insert()**, **index()**) and functions (**len()**, **sorted()**, etc.), **del** instruction, iterating lists with the **for** loop, initializing, **in** and **not in** operators, list comprehension, copying and cloning
- lists in lists: matrices and cubes
- tuples: indexing, slicing, building, immutability
- tuples vs. lists: similarities and differences, lists inside tuples and tuples inside lists
- dictionaries: building, indexing, adding and removing keys, iterating through dictionaries as well as their keys and values, checking key existence, **keys()**, **items()** and **values()** methods
- strings in detail: escaping using the **** character, quotes and apostrophes inside strings, multi-line strings, basic string functions.

Exam block #4: Functions and Exceptions (28%)

Objectives covered by the block (8 exam items)

- **PCEP 4.1 Decompose the code using functions**
  defining and invoking user-defined functions and generators; the return keyword, returning results, the None keyword, recursion
- **PCEP 4.2 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), the global keyword
- **PCEP 4.3 Python Built-In Exceptions Hierarchy**
  BaseException, Exception, SystemExit, KeyboardInterrupt, abstractive exceptions, ArithmeticError, LookupError along with IndexError and KeyError; TypeError and ValueError exceptions, the AssertError exception along with the assert keyword
- **PCEP 4.4 Basics of Python Exception Handling**
  try-except, try-except Exception, ordering the except branches, propagating exceptions through function boundaries; delegating responsibility for handling exceptions

Exam block #5: Functions (20%)

Objectives covered by the block (6 exam items)

- defining and invoking your own functions and generators
- **return** and **yield** keywords, returning results,
- the **None** keyword,
- recursion
- parameters vs. arguments,
- positional keyword and mixed argument passing,
- default parameter values
- converting generator objects into lists using the **list()** function
- name scopes, name hiding (shadowing), the **global** keyword

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

Last updated: February 2, 2022
Aligned with PCEP 30-02