EXAM PCAP SYLLABUS:

**SECTION 1: MODULES AND PACKAGES (12%)**

Objectives covered by the block (6 exam items)

***PCAP-31-03 1.1 – Import and use modules and packages***

* import variants: import, from import, import as, import \*
* advanced qualifying for nested modules
* the *dir()* function
* the *sys.path* variable

***PCAP-31-03 1.2 – Perform evaluations using the math module***

* functions: ceil(), floor(), trunc(), factorial(), hypot(), sqrt()

***PCAP-31-03 1.3 – Generate random values using the random module***

* functions: random(), seed(), choice(), sample()

***PCAP-31-03 1.4 – Discover host platform properties using the platform module***

* functions: platform(), machine(), processor(), system(), version(), python\_implementation(), python\_version\_tuple()

***PCAP-31-03 1.5 – Create and use user-defined modules and packages***

* idea and rationale;
* the *\_\_pycache\_\_* directory
* the *\_\_name\_\_* variable
* public and private variables
* the *\_\_init\_\_.py* file
* searching for/through modules/packages
* nested packages vs. directory trees

**SECTION 2: EXCEPTIONS (14%)**

Objectives covered by the block (5 exam items)

***PCAP-31-03 2.1 – Handle errors using Python-defined exceptions***

* except, except:-except, except:-else:, except (e1, e2)
* the hierarchy of exceptions
* raise, raise ex
* assert
* event classes
* except E as e
* the *arg* property

***PCAP-31-02 2.2 – Extend the Python exceptions hierarchy with self-defined exceptions***

* self-defined exceptions
* defining and using self-defined exceptions

**SECTION 3: STRINGS (18%)**

Objectives covered by the block (8 exam items)

***PCAP-31-03 3.1 – Understand machine representation of characters***

* encoding standards: ASCII, UNICODE, UTF-8, code points, escape sequences

***PCAP-31-03 3.2 – Operate on strings***

* functions: *ord()*, *chr()*
* indexing, slicing, immutability
* iterating through strings, concatenating, multiplying, comparing (against strings and numbers)
* operators: *in*, *not in*

***PCAP-31-03 3.3 – Employ built-in string methods***

* methods: *.isxxx()*, *.join()*, *.split()*, *.sort()*, *sorted()*, *.index()*, *.find()*, *.rfind()*

**SECTION 4: OBJECT ORIENTED PROGRAMMING (34%)**

Objectives covered by the block (12 exam items)

***PCAP-31-03 4.1 – Understand the Object-Oriented approach***

* ideas and notions: class, object, property, method, encapsulation, inheritance, superclass, subclass, identifying class components

***PCEP-31-03 4.2 – Employ class and object properties***

* instance vs. class variables: declarations and initializations
* the *\_\_dict\_\_* property (objects vs. classes)
* private components (instances vs. classes)
* name mangling

***PCAP-31-03 4.3 – Equip a class with methods***

* declaring and using methods
* the *self* parameter

***PCAP-31-03 4.4 – Discover the class structure***

* introspection and the *hasattr()* function (objects vs classes)
* properties: *\_\_name\_\_*, *\_\_module\_\_* , *\_\_bases\_\_*

***PCAP-31-03 4.5 – Build a class hierarchy using inheritance***

* single and multiple inheritance
* the *isinstance()* function
* overriding
* operators:
* not is, *is*
* polymorphism
* overriding the *\_\_str\_\_()* method
* diamonds

***PCAP-31-03 4.6 – Construct and initialize objects***

* declaring and invoking constructors

**SECTION 5: MISCELLANEOUS (22%)**

***Scope: List Comprehensions, Lambdas, Closures, and I/O Operations***

Objectives covered by the block (9 exam items)

***PCAP-31-03 5.1 – Build complex lists using list comprehension***

* list comprehensions: the *if* operator, nested comprehensions

***PCAP-31-03 5.2 – Embed lambda functions into the code***

* lambdas: defining and using lambdas
* self-defined functions taking lambdas as arguments
* functions: *map()*, *filter()*

***PCAP-31-03 5.3 – Define and use closures***

* closures: meaning and rationale
* defining and using closures

***PCAP-31-03 5.4 – Understand basic Input/Output terminology***

* I/O modes
* predefined streams
* handles vs. streams
* text vs. binary modes

***PCAP-31-03 5.5 – Perform Input/Output operations***

* the *open()* function
* the *errno* variable and its values
* functions: *close()*, *.read()*, *.write()*, *.readline()*, *readlines()*
* using bytearray as input/output buffer