Certified Associate in Python Programming (PCAP)

Duration: 32 Hours (4 Days)

Overview

The Certified Associate in Python Programming (PCAP) course is a comprehensive program designed to equip learners with a robust understanding of Python, one of the most popular programming languages. Aimed at both beginners and those looking to formalize their skills, this course offers a deep dive into Python's essential concepts and constructs through four detailed Modules.Module 1 focuses on the foundational elements like Functions, Modules, and Packages, including how to create your own, manage them using PIP, and work with File handling. Module 2 enhances your knowledge of Python's data structures, such as strings, lists, tuples, and sets, along with Error and exception handling, and introduces List comprehension. Module 3 delves into Object-Oriented Programming, covering Classes, Objects, Methods, Inheritance, and Polymorphism. Lastly, Module 4 touches upon advanced topics like Generators, Iterators, and various Modules for System operations and Time management, as well as best practices in Testing and code quality with Pylint.Earning the PCAP certification validates a candidate's proficiency in Python, which is aligned with the PCAP 31-03 exam objectives. This credential not only enhances a learner's resume but also bolsters their ability to tackle real-world programming challenges with Python's powerful capabilities.

Audience Profile

- The Certified Associate in Python Programming course offers comprehensive training in Python essentials, suitable for beginners and intermediate programmers.
- Target Audience for the Certified Associate in Python Programming Course:
- Aspiring software developers
- Computer science students
- Data analysis enthusiasts
- Entry-level programmers
- IT professionals looking to expand their skill set
- Automation engineers
- Quality assurance specialists
- System administrators
- Academic researchers
- Hobbyists interested in learning programming
- Technical product managers
- Professionals in tech roles seeking to learn a new scripting language

Course Syllabus

PCAP: Certified Associate in Python Programming

This course is the second in a 2-course series that will prepare you for the PCAP: Certified Associate in Python Programming certification exam. The course picks up where PCEP course leaves off.

Module 1: Modules and Packages

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

2 – Perform evaluations using the math module

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

3 – Generate random values using the random module

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

4 – Discover host platform properties using the platform module

- functions: platform(), machine(), processor(), system(), version(),
- python_implementation(), python_version_tuple()

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

Module 2 Exceptions

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

2 – Extend the Python exceptions hierarchy with self-defined

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

Module 3 Strings

1 – Understand machine representation of characters

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

2 – Operate on strings

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

3 – Employ built-in string methods

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

Module 4 : Object-Oriented Programming

1 – Understand the Object-Oriented approach

- Class
- Object property, method
- Encapsulation
- Inheritance
- Superclass
- Subclass
- identifying class components

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

3 – Equip a class with methods

- declaring and using methods
- the self parameter

4 – Discover the class structure

- introspection and the hasattr() function (objects vs classes)
- properties: __name__, __module__ , __bases__

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

6 – Construct and initialize objects

- declaring and invoking constructors

Module 5 Miscellaneous

1 – Build complex lists using list comprehension

- list comprehensions: the if operator, nested comprehensions

2 – Embed lambda functions into the code

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

3 – Define and use closures

- closures: meaning and rationale
- defining and using closures

4 – Understand basic Input/Output terminology

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

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