Python Developer Associate Certification Study Guide

Please use this study guide to create your certification self-study plan. We've included the objectives you should meet for each assessed competency, with links to relevant practice assessments.

Exam PDA101: Core Python, Building with Python, and Developer Workflow and Tooling

1.1 Use core Python syntax and built-in data structures to write functional programs

- Write and execute programs using variables, operators, conditionals, and loops.
- Manipulate strings using built-in methods, f-strings, and formatting.
- Create and manipulate lists, tuples, sets, and dictionaries.

1.2 Write modular, reusable code using Python functions and functional constructs

- Define and call functions using positional arguments, keyword arguments, and defaults.
- Write lambda functions for single-expression operations.
- Use iterators, generators, and comprehensions including list, dict, and generator expressions, and yield.
- Write docstrings and documentation for functions and classes.

1.3 Use Python's standard library and built-in modules effectively

- Use os for file system and environment operations.
- Use Counter, defaultdict, and dataclasses for structured data.
- Work with dates and times using datetime, timedelta, and dateutil.
- Write and apply regular expressions for pattern matching, grouping, and substitution.

1.4 Apply Python control structures for robust and readable code

- Use decorators from the standard library.
- Use context managers with statements for file handling and resource management.
- Handle errors using try/except blocks and raise exceptions appropriately.

Python Developer Associate Certification Study Guide

2.1 Write object-oriented Python programs using classes and inheritance

- Define classes with __init__, attributes, and methods.
- Implement single inheritance and method overriding.
- Use standard dunder methods for operator overloading.

2.2 Read and evaluate code written by others or generated by tools

- Read and interpret unfamiliar code and understand its intent.
- Read tracebacks and use print statements or a debugger to identify errors.
- Review code for correctness, edge case handling, and style consistency.

2.3 Use pandas for basic data handling tasks

- Load data from CSV files using read_csv.
- Inspect and manipulate DataFrames.

2.4 Use AI coding tools effectively and critically

- Generate and modify code for discrete, well-scoped tasks using AI tools.
- Accept, reject, or edit suggestions from AI tools, including recognising when AI output requires manual correction.

3.1 Navigate and operate a Linux/Unix environment using the command line

- Navigate the file system and manage files using pwd, ls, cd, cp, mv, and rm.
- Process and inspect text files using cat, grep, head, tail, and cut.
- Combine commands using pipes, redirects, and wildcards.

3.2 Write and execute shell scripts

- Write shell scripts using variables, loops, and arguments.