Microsoft Python Certification Overview

Topic	Details
Types of Jobs	Python has become one of the most used programming languages and the Python certification is one of the most sought after requirements in the entire programming landscape. Although Python can be used for little projects, this programming language is used by many tech giants, such as Microsoft, Facebook, Netflix, Google, and even NASA. Hiring preference is given to those having a Python certification**.**
About the Exam	Candidates for this exam should be able to recognize and write syntactically correct Python code, recognize data types supported by Python, and be able to recognize and write Python code that will logically solve a given problem.
Exam Details	**Exam Details** • **Required exam**: 98–381 • **Number of questions** 40 questions • **Types of questions:** Question types vary. • **Length of test:** 45 minutes • **Passing score:** Any score of 700 or greater is a pass. Any score below 700 is a fail. **Recommended experience** • Candidates are expected to have had, at a minimum, instruction and/or hands-on experience of approximately 100 hours with the Python programming language, be familiar with its features and capabilities, and understand how to write, debug, and maintain well-formed, well-documented Python code.

Microsoft Python Certification Overview (Cont.)

Contain Domains	**Domain**	**% of Examination**	**Objectives** *Please note that the questions may test on but will not be limited to the topics described in the bulleted text.*
	Perform Operations Using Data Types and Operators	20–25%	• Evaluate an expression to identify the data type Python will assign to each variable. o Identify str, int, float, and bool data types • Perform data and data type operations. o Convert from one data type to another type; construct data structures; perform indexing and slicing operations. • Determine the sequence of execution based on operator precedence. o Assignment; Comparison; Logical; Arithmetic; Identity (is); Containment (in) • Select the appropriate operator to achieve the intended result. o Assignment; Comparison; Logical; Arithmetic; Identity (is); Containment (in)
	Control Flow With Decisions and Loops	20–25%	• Construct and analyze code segments that use branching statements. o if; elif; else; nested and compound conditional expressions • Construct and analyze code segments that perform iteration. o while; for; break; continue; pass; nested loops and loops that include compound conditional expressions
	Perform Input and Output Operations	20–25%	• Construct and analyze code segments that perform file input and output operations. o Open; close; read; write; append; check existence; delete; with statement • Construct and analyze code segments that perform console input and output operations. o Read input from console; print formatted text; use of command line arguments.
	Document and Structure Code	15–20%	• Document code segments using comments and documentation strings. o Use indentation, white space, comments, and documentation strings; generate documentation by using pydoc. • Construct and analyze code segments that include function definitions. o Call signatures; default values; return; def; pass
	Perform Troubleshooting and Error Handling	5–10%	• Analyze, detect, and fix code segments that have errors. o Syntax errors; logic errors; runtime errors • Analyze and construct code segments that handle exceptions. o Try; except; else; finally; raise
	Perform Operations Using Modules and Tools	1–5%	• Perform basic operations using built-in modules. o Math; datetime; io; sys; os; os.path; random • Solve complex computing problems by using built-in modules. o Math; datetime; random
	Total	**100%**	