

Python

1. Operations using Data Types and Operators

- 1.1 Evaluate expressions to identify the data types Python assigns to variables
 - str, int, float, and bool
- 1.2 Perform data and data type operations
 - Data type conversion, indexing, slicing, construct data structures
- 1.3 Determine the sequence of execution based on operator precedence
 - Assignment, comparison, logical, arithmetic, identity (is), containment (in)
- 1.4 Select operators to achieve the intended results
 - Assignment, comparison, logical, arithmetic, identity (is), containment (in)

2. Flow Control with Decisions and Loops

- 2.1 Construct and analyze code segments that use branching statements
 - if, elif, else, nested and compound conditional expressions
- 2.2 Construct and analyze code segments that perform iteration
 - while, for, break, continue, pass, nested loops, loops that include compound conditional expressions

3. Input and Output Operations

- 3.1 Construct and analyze code segments that perform file input and output operations
 - open, close, read, write, append, check existence, delete, with statement
- 3.2 Construct and analyze code segments that perform console input and output operations
 - Read input from console, print formatted text (string.format() method, f-String method), use command-line arguments

4. Code Documentation and Structure

- 4.1 Document code segments
 - Use indentation, white space, comments, and documentation strings; generate documentation by using pydoc
- 4.2 Construct and analyze code segments that include function definitions
 - Call signatures, default values, return, def, pass

5. Troubleshooting and Error Handling



IT SPECIALIST EXAM OBJECTIVES

5.1 Analyze, detect, and fix code segments that have errors

- Syntax errors, logic errors, runtime errors

5.2 Analyze and construct code segments that handle exceptions

- try, except, else, finally, raise

5.3 Perform unit testing

- unittest, functions and methods

6. Operations using Modules and Tools

6.1 Perform basic operations by using built-in modules

- math, datetime, io, sys, os, os.path, random

6.2 Solve complex computing problems by using built-in modules

- math, datetime, random



INFORMATION
TECHNOLOGY
SPECIALIST