

Python Institute Certification Objectives

The test candidate should demonstrate the sufficient knowledge of the following concepts:

- 1.** The fundamentals of computer programming, i.e. how the computer works, how the program is executed, how the programming language is defined and constructed, what the difference is between compilation and interpretation, what Python is, how it is positioned among other programming languages, and what distinguishes the different versions of Python;
- 2.** The basic methods of formatting and outputting data offered by Python, together with the primary kinds of data and numerical operators, their mutual relations and bindings; the concept of variables and variable naming conventions; the assignment operator, the rules governing the building of expressions; the inputting and converting of data;
- 3.** Boolean values to compare difference values and control the execution paths using the *if* and *if-else* instructions; the utilization of loops (*while* and *for*) and how to control their behavior using the *break* and *continue* instructions; the difference between logical and bitwise operations; the concept of lists and list processing, including the iteration provided by the *for* loop, and slicing; the idea of multi-dimensional arrays;
- 4.** The defining and using of functions – their rationale, purpose, conventions, and traps; the concept of passing arguments in different ways and setting their default values, along with the mechanisms of returning the function's results; name scope issues; new data aggregates: tuples and dictionaries, and their role in data processing;

5. Python modules: their rationale, function, how to import them in different ways, and present the content of some standard modules provided by Python; the way in which modules are coupled together to make packages; the concept of an exception and Python's implementation of exceptions, including the *try-except* instruction, with its applications, and the *raise* instruction; strings and their specific methods, together with their similarities and differences compared to lists;

6. The fundamentals of OOP (Object Oriented Programming) and the way they are adopted in Python, showing the difference between OOP and the classical, procedural approach; the standard objective features: inheritance, abstraction, encapsulation, and polymorphism, along with Python-specific issues like instance vs. class variables, and Python's implementation of inheritance; objective nature of exceptions; Python's generators (the *yield* instruction) and closures (the *lambda* keyword); the means Python developers can use to process (create, read, and write) files.