

# PCAP | Programming Essentials in Python

## Scope and Sequence

Last updated December 19, 2017

### Scope and Sequence – Table of Contents

1. Target Audience
2. Prerequisites
3. Target Certification
4. Curriculum Description
5. Curriculum Objectives
6. Course Outline
7. Minimum System Requirements
8. Industry certification

## Target Audience

The *PCAP | Programming Essentials in Python* curriculum is designed for students with little or no prior knowledge of programming, i.e., students of secondary school, university, vocational school, or simply anyone interested in learning programming. The only preliminary requirement is the ability to use a personal computer and a very basic knowledge of mathematics.

## Prerequisites

There are no prerequisites for this course.

## Target Certification

The *PCAP | Programming Essentials in Python* curriculum helps students prepare for the [PCAP | Python Certified Associate Programmer certification](#) exam. *PCAP | Python Certified Associate Programmer* is a professional certification that measures the student's ability to accomplish coding tasks related to the basics of programming in the Python language and the fundamental notions and techniques used in object-oriented programming.

## Curriculum Description

The *PCAP | Programming Essentials in Python* course covers all the basics of programming in Python, as well as general computer programming concepts and techniques. The course also familiarizes the student with the object-oriented approach. The course is broken down into five modules.

Each student has access to hands-on practice materials, quizzes, and assessments to learn how to utilize the skills and knowledge gained on the course and interact with some real-life programming tasks and situations.

## Curriculum Objectives

The aim of the course is to familiarize the student with general computer programming concepts like conditional execution, loops, Python programming language syntax, semantics, and the runtime environment, as well as with general coding techniques and object-oriented programming.

Completing the course ensures that the student is fully acquainted with all the primary means provided by Python 3 to enable her/him to start her/his own studies, and to open a path to the developer's career.

Module objectives:

### Module 0 (optional)

Familiarize the student with the fundamentals of computer programming: 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. Demonstrate a simple way to download, install, and run the Python environment on a personal computer.

### Module 1

Familiarize the student with the basic methods offered by Python of formatting and outputting data, together with the primary kinds of data and numerical operators, their mutual relationships and binding. Introduce the concept of variables and variable naming conventions. Present the assignment operator, along with the rules governing the building of expressions. Introduce the inputting and converting of data.

## Module 2

Familiarize the student with the concept of Boolean values, in order to compare difference values and to control the execution paths using the `if` and `if-else` instructions. Introduce the utilization of loops (`while` and `for`) and how to control their behavior using the `break` and `continue` instructions. Present the difference between logical and bitwise operations. Acquaint the student with the concept of lists and list processing, including the iteration provided by the `for` loop, and slicing. Explain the idea of multi-dimensional arrays.

## Module 3

Acquaint the student with the defining and using of functions – their rationale, purpose, conventions, and traps. Present the concept of passing arguments in different ways and setting their default values, along with the mechanisms of returning the function’s results. Explain name scope issues. Introduce new data aggregates – tuples and dictionaries – and show their role in data processing.

## Module 4

Familiarize the student with Python modules: their rationale, function, how to import them in different ways, and present the contents of some standard modules provided by Python. Present the way in which modules are coupled together to make packages. Acquaint the student with the concept of an exception and Python’s implementation of it, including the `try-except` instruction, with its applications, and the `raise` instruction. Introduce strings and their specific methods, together with their similarities and differences compared to lists.

## Module 5

Acquaint the student with 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. Present 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. Exceptions are discussed again in a more detailed way, showing their objective nature. Familiarize the student with Python’s generators (the `yield` instruction) and closures (the `lambda` keyword). Demonstrate the means Python developers can use to process (create, read, and write) files.

## Course Outline

Learning Module	PCAP   Python Certified Associate Programmer certification objectives covered
<b>1 – Basics I</b>	<ul style="list-style-type: none"> <li>the <code>print()</code> function, formatting the output</li> <li>literals (integers, floats, strings, Boolean values)</li> <li>operators and expressions, arithmetic operators, operators and their priorities, operators and their bindings</li> <li>variables, naming and assigning variables, shortcut operators</li> <li>comments</li> <li>output vs. input, inputting data with the <code>input()</code> function</li> <li>converting strings into numbers, simple interactive programs, string operators, converting numbers into strings</li> </ul>
<b>2 – Basics II</b>	<ul style="list-style-type: none"> <li>asking questions and receiving answers, relational operators</li> <li>conditions and conditional execution, the <code>if</code> statements, the <code>if-else</code> statements, the <code>elif</code> clause</li> <li>loops (<code>while</code>, <code>for</code>, <code>break</code>, <code>continue</code>)</li> </ul>

	<ul style="list-style-type: none"><li>• computer logic and its operators, logical values vs. single bits</li><li>• bitwise operators, dealing with single bits</li><li>• creating and using lists, removing elements from a list, list methods (methods vs. functions), adding elements to a list, lists in action</li><li>• sorting lists (the bubble sort algorithm)</li><li>• storing lists, slices, the <code>in</code> and <code>not in</code> operators</li><li>• lists in advanced applications (lists within lists, list comprehension, matrices, 3<sup>rd</sup> dimension)</li></ul>
<b>3 – Basics III</b>	<ul style="list-style-type: none"><li>• designing and writing functions</li><li>• parametrized functions, defining and using function parameters, shadowing, positional arguments, keyword arguments, mixed arguments, sorting parameter default values</li><li>• returning a result from a function (the <code>return</code> statement, returning a value, the <code>None</code> value, returning the non-<code>None</code> value, argument vs. parameter compatibility, a list as a function's result)</li><li>• functions and scopes, global variables, interaction of parameters with their arguments</li><li>• recursion</li><li>• tuples and dictionaries (sequence types and mutability, creating and using tuples, creating and using dictionaries)</li></ul>
<b>4 – Intermediate I</b>	<ul style="list-style-type: none"><li>• using and importing modules</li><li>• working with standard modules</li><li>• functions from the <code>math</code> module, functions from the <code>random</code> module, functions from the <code>platform</code> module</li><li>• modules and packages</li><li>• errors, failures, exceptions</li><li>• characters and strings vs. computers, the nature of strings in Python, string methods, strings in action (comparing strings, sorting strings, strings vs. numbers)</li><li>• simple programs</li></ul>
<b>5 – Intermediate II</b>	<ul style="list-style-type: none"><li>• the basic concepts of object programming, classes</li><li>• a stack</li><li>• properties (instance variables, class variables, checking an attribute's existence)</li><li>• methods (the inner life of classes and objects, reflection and introspection, classes and methods in detail)</li><li>• inheritance (finding properties and methods, building a hierarchy of classes, inheritance vs. composition, single inheritance vs. multiple inheritance)</li><li>• exceptions – advanced topics, creating and using exceptions</li><li>• generators and closures (the <code>yield</code> statement, building generators, list comprehensions – advanced topics, the <code>lambda</code> function)</li><li>• processing files (accessing files from Python code, file names, file streams, file handles, opening the streams, selecting text and binary modes, pre-opened streams, closing streams, diagnosing stream problems)</li><li>• working with real files (dealing with text files, working with binary files, stream – reading and writing bytes, copying files)</li></ul>

### Minimum System Requirements

The course content modules, labs, quizzes and assessments can be accessed online through any Internet browser. For the best learning experience, we recommend using the most recent versions of Mozilla Firefox, Internet Explorer/Microsoft Edge, or Google Chrome.

### Industry certification

The course curriculum helps students prepare for the Python Institute [PCAP | Python Certified Associate Programmer certification](#).

A Statement of Achievement will be issued to participants who successfully complete the *PCAP | Programming Essentials in Python* course. The Statement of Achievement will acknowledge that the individual has completed the course and is now ready to attempt the qualification *PCAP | Python Certified Associate Programmer Certification*, taken through Pearson VUE computer-based testing, at a 51% discount.

To receive the Statement of Achievement, instructors must mark the student as having successfully passed the course.

For additional information about the *Python Institute PCAP | Python Certified Associate Programmer certification*, please visit [www.pythoninstitute.org/certification](http://www.pythoninstitute.org/certification).