

PCAP: Programming Essentials in Python (Python Essentials)

Scope & Sequence

PCAP: Programming Essentials in Python
(aka **Python Essentials**)
Developed by **OpenEDG Python Institute**

PCAP: Programming Essentials in Python

Scope and Sequence

Table of contents

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

Target Audience

PCAP: Programming Essentials in Python (short: **Python Essentials**) is designed for students with little or no prior knowledge of programming; 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 very basic knowledge in mathematics.

Prerequisites

There are no prerequisites for this course.

Curriculum Description

The *PCAP: Programming Essentials in Python* course covers all the basics of programming in Python 3, 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 two parts, each divided into four modules. Students have 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

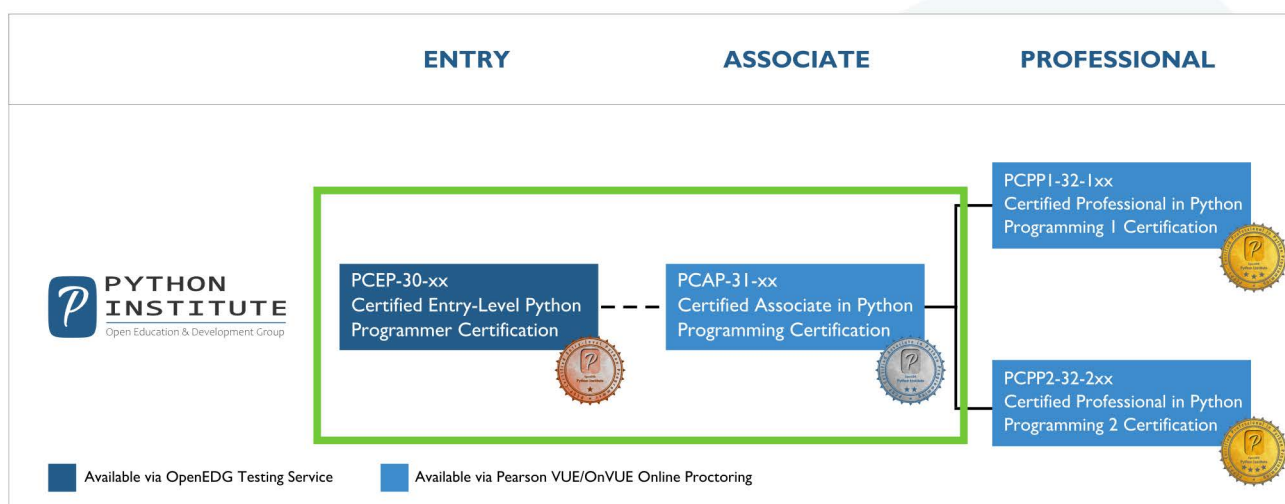
The aim of the course is to familiarize students with general computer programming concepts like **conditional execution**, **functions**, **loops**, Python programming language **syntax**, **semantics**, and the **runtime environment**, as well as with general **coding techniques** and **object-oriented programming**. Completing the course ensures that the student is fully acquainted with all the primary means provided by Python 3 to enable them to start their own studies, and to open a path to the developer's career.

Industry Certification

The *PCAP: Programming Essentials in Python* curriculum helps students prepare for the *PCEP – Certified Entry-Level Python Programmer* (PE1: Modules 1-4) and *PCAP – Certified Associate in Python Programming* (PE2: Modules 1-4) certification exams.

PCEP – Certified Entry-Level Python Programmer certification is a professional credential that measures the student's ability to accomplish coding tasks related to the essentials of programming in the Python language.

PCAP – Certified Associate in Python Programming measures the skills and knowledge related to the more advanced aspects of Python programming, including the fundamental notions and techniques used in the object-oriented approach (OOP).



A **Statement of Achievement** will be issued to participants who successfully complete the instructor-led version of 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 – Certified Associate in Python Programming* certification, taken through Pearson VUE computer-based testing, at a 50% discount.

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

For additional information about the *PCEP – Certified Entry-Level Python Programmer* and *PCAP – Certified Associate in Python Programming* certifications, please visit pythoninstitute.org/certification.

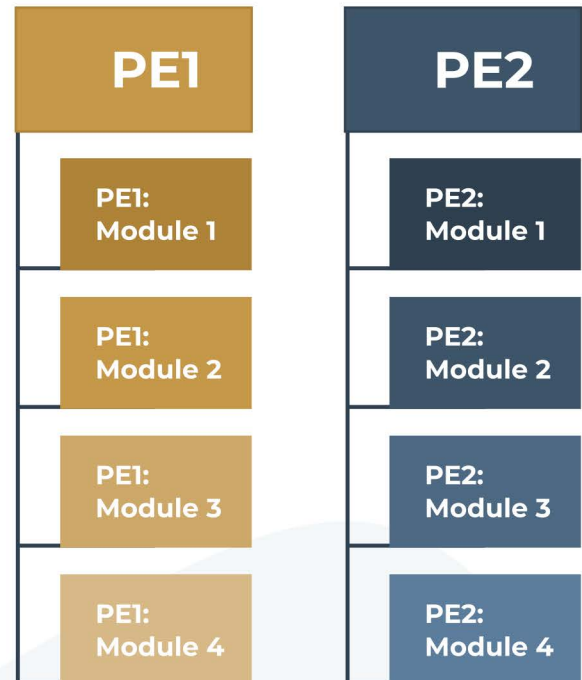
Course Structure

The *PCAP: Programming Essentials in Python (2.0)* course is divided into two parts. Each part can be taught as an independent mini-course over a semester.

Python Essentials – Part 1 (PE1) is aligned with PCEP certification, while Python Essentials – Part 2 (PE2) is aligned with PCAP certification.

Each module concludes with a brief quiz and a module test. Additionally, each part (PE1 and PE2) ends with a Summary Test, which includes all the most important questions covered in modules 1 through 4.

PCAP: Programming Essentials in Python (2.0)



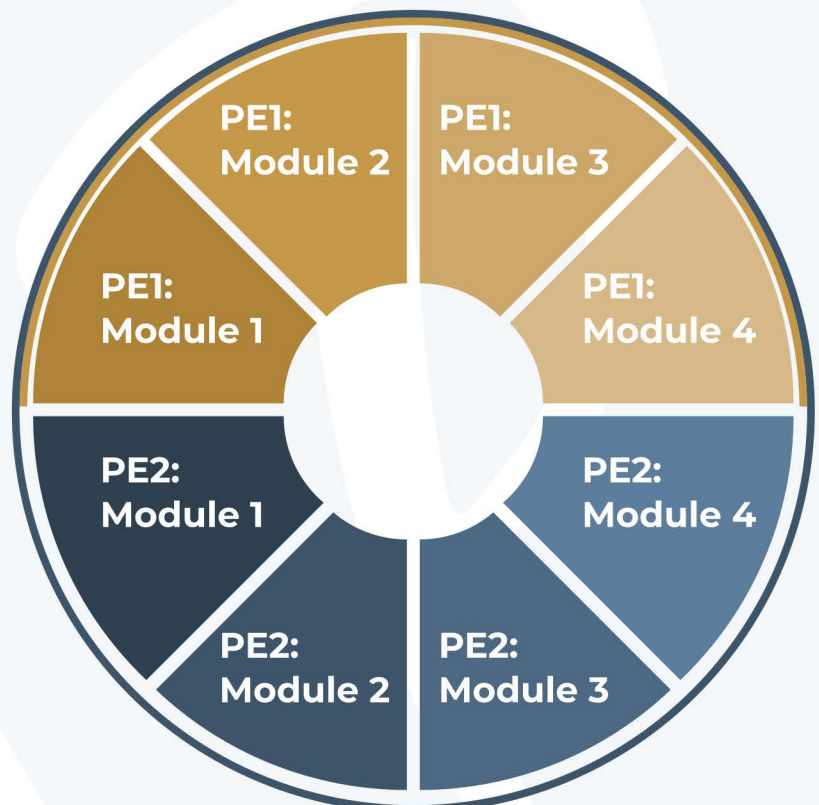
Python Essentials 1, modules 1, 2, 3 and 4 will prepare you for:

PCEP – Certified Entry-Level Python Programmer certification exam



Python Essentials 2, modules 1, 2, 3 and 4 will prepare you for:

PCAP – Certified Associate in Python Programming certification exam



Course Objectives

Python Essentials – Part 1 (PE1)

PE1: Module 1

After completing PE1 Module 1, the student will:

- ✓ have a basic knowledge of computer programming and software development;
- ✓ understand the fundamental programming concepts, such as: compiler, interpreter, source code, machine code, IDE;
- ✓ have an orientation in Python's development history, its main traits and features;
- ✓ gain skills allowing her/him to install and configure basic development tools as well as code, and run the very first Python program.

PE1: Module 2

After completing PE1 Module 2, the student will:

- ✓ gain skills enabling her/him to create, edit and run Python source files using IDLE;
- ✓ have some knowledge of Python's numeral literals, their syntax, types and formats;
- ✓ have an orientation in issues related to Python arithmetic operators and expressions; gain the ability to name, create, initialize and modify variables;
- ✓ have skills that enable her/him to perform basic input/output operations in a Python program.

PE1: Module 3

After completing PE1 Module 3, the student will:

- ✓ know basic features of the Boolean data type;
- ✓ gain skills to work with relational operators in Python;
- ✓ have the ability to effectively use the control statements `if`, `if-else` and `if-elif-else`;
- ✓ understand the role of a loop and be able to use the control statements `while` and `for`;
- ✓ have an orientation in bitwise operations in Python;
- ✓ know the role of lists and be able to operate with them to perform actions including indexing, slicing and content manipulation;
- ✓ understand how the bubble-sort algorithm works;
- ✓ have a knowledge of multidimensional lists in Python.

PE1: Module 4

After completing PE1 Module 4, the student will:

- ✓ understand the concept of functions and be able to code and invoke her/his own functions;
- ✓ have an orientation of the main features of structural programming;
- ✓ have some knowledge of name scopes and be able to distinguish global and local variables, as well as understand how name shadowing works;
- ✓ understand the principles of tuples including the immutability notion;
- ✓ know the role of dictionaries and be able to use them effectively in appropriate circumstances;
- ✓ be ready to take the Part 1 Summary Test, and attempt the qualification *PCEP – Certified Entry-Level Python Programmer* from the OpenEDG Python Institute.

Graduate profile

A student who has completed the PE1 course will have acquired:

- ✓ an ability to design, develop and improve simple computer programs coded in Python;
 - ✓ a knowledge suitable to start learning another programming language
 - ✓ sufficient competence to take the PCEP exam and obtain the corresponding certificate;
 - ✓ the skills needed to take part in the next Python Institute course PE2;
 - ✓ experience allowing her/him to take a job as a junior tester;
 - ✓ the possibility to continue her/his personal development through self-education and self-improvement.
-

Python Essentials – Part 2 (PE2)

PE2: Module 1

After completing PE2 Module 1, the student will:

- ✓ understand the role of the Python module and know the available ways of importing modules into her/his own code/namespace;
- ✓ gain knowledge of selected useful standard Python modules;
- ✓ have an orientation in package purposes as well as be able to create her/his own packages;
- ✓ know the main function of PIP and be able to use it in order to install and uninstall ready-to-use packages from PyPI.

PE2: Module 2

After completing PE2 Module 2, the student will:

- ✓ know how characters are coded and stored inside the computer's memory, distinguish most known coding standards;
- ✓ gain knowledge of Python's sequences and know the differences between strings and lists;
- ✓ be able to effectively use selected lists and string methods;
- ✓ have an orientation of Python's way of identifying and handling runtime errors;
- ✓ understand the purpose of the control statements **try**, **except** and **raise**;
- ✓ understand Python exception hierarchies.

PE2: Module 3

After completing PE2 Module 3, the student will:

- ✓ understand the fundamental concepts of object programming like class, object, property, method, inheritance and polymorphism;
- ✓ have an orientation in the differences between procedural and object approaches, as well as being oriented when both of the techniques reveal their pros and cons;
- ✓ be able to build her/his own classes, objects, properties and methods;
- ✓ be able to use inheritance and polymorphism in her/his inheritance path;
- ✓ understand the objective nature of Python exceptions.

PE2: Module 4

After completing PE2 Module 4, the student will:

- ✓ gain the ability to understand the concepts of generators, iterators and closures as well as be able to use them in adequate applications;
- ✓ know how Python accesses physical file-system resources, understand file open modes and perform basic input/output operations in relation to text and binary files;
- ✓ gain an ability to manipulate date and time, work with a calendar, and create directory structures using Python;
- ✓ be ready to take the Part 2 Summary Test, and the Final Test;
- ✓ be prepared to attempt the qualification *PCAP – Certified Associate in Python Programming* from the OpenEDG Python Institute.

Graduate profile

A student who has completed the PE2 course will have acquired:

- ✓ an ability to design, develop and improve multi-module computer applications coded in Python;
 - ✓ an ability to analyze and model real-life problems in OOP categories;
 - ✓ sufficient competences to take the PCAP exam and obtain the corresponding certificate;
 - ✓ experience allowing her/him to take a job as a junior developer;
 - ✓ sufficient skills to create and develop her/his own programming portfolio;
 - ✓ the potential to use Python in everyday life applications including DIY activities.
-

Course Outline

Learning Module	Course Syllabus
Part 1 – Basics Module 1 <i>Introduction to Python and Computer Programming</i>	<ul style="list-style-type: none">- Python as a modern, universal and growing programming language;- Python versions and language development;- Brief review of tools and environments needed to start programming in Python.
Part 1 – Basics Module 2 <i>Data Types, Variables, Basic Input-Output Operations, Basic Operators</i>	<ul style="list-style-type: none">- How to write and run the very first Python program;- Python literals;- Python operators and expressions;- Variables – naming and using;- Basic input/output operations in Python.
Part 1 – Basics Module 3 <i>Boolean Values, Conditional Execution, Loops, Lists and List Processing, Logic and Bitwise Operations</i>	<ul style="list-style-type: none">- Boolean data type;- Relational operators in Python;- Decision making in Python: if, if-else, if-elif-else;- Repeating code execution using loops: while and for;- Logic and bitwise operations in Python;- Lists in Python: constructing, indexing, slicing and content manipulation;- How to sort a list using a bubble-sort algorithm;- Multidimensional lists and their applications.
Part 1 – Basics Module 4 <i>Functions, Tuples, Dictionaries, and Data Processing</i>	<ul style="list-style-type: none">- Code structuring and the concept of functions;- Function invocation and returning a result from a function;- Name scopes and variable shadowing;- Tuples – purpose, constructing and using;- Dictionaries – purpose, constructing and using.

Course Outline

Learning Module	Course Syllabus
Part 2 – Intermediate Module 1 <i>Modules, Packages and PIP</i>	<ul style="list-style-type: none">- What is a module and why do we need it?- Importing and using modules;- Review of some useful Python modules;- What is a package and how does it differ from a module?- Constructing and using packages;- PIP – the Swiss army knife for package maintenance.
Part 2 – Intermediate Module 2 <i>Exceptions, Strings, String and List Methods</i>	<ul style="list-style-type: none">- Characters, strings and coding standards;- Strings vs. lists – similarities and differences;- List methods;- String methods;- Python's way of handling runtime errors;- Controlling the flow of errors using try and except;- Hierarchy of exceptions.
Part 2 – Intermediate Module 3 <i>Object Oriented Programming in Python</i>	<ul style="list-style-type: none">- Basic concepts of object programming;- From procedural to object approach – motivations and profits;- Classes, objects, properties and methods;- Inheritance and polymorphism;- Exception as an object.
Part 2 – Intermediate Module 4 <i>Miscellaneous</i>	<ul style="list-style-type: none">- Generators, iterators and closures;- Working with filesystem, directory trees and files;- Selected Python Standard Library modules (os, date, datetime, calendar)

Minimum System Requirements

The course can be accessed online through any Internet browser, on computers with Linux, Windows, or Mac OS. The minimum equipment required for this course is:

- a computer with an Internet browser and active Internet connection, equipped with an IDE, or
- a computer with an Internet browser and active Internet connection.

The first option requires having the Python 3 standard installation on your computer. A copy of Python 3 can be downloaded from <https://www.python.org/downloads>. The installation contains a software application called IDLE (Integrated Development and Learning Environment), which will enable you to execute simple Python commands and see the effects of executing your programs.

Full information about how to get Python, how to install it, and how to use it is available in Module 1 of the PCAP: Programming Essentials in Python course.

The second option does not require the installation of any software applications – it is possible to use a dedicated, interactive on-line programming environment (Edube Interactive) that allows Python code to be run in an Internet browser. Edube Interactive is a tool integrated with the course, which can be used as a browser-based Python sandbox that allows you to test the code discussed throughout the course, as well as an interpreter that enables you to launch, perform, and test lab exercises.

Recommended equipment and technical requirements to use Edube Interactive:

- a desktop computer or a laptop (recommended: a desktop computer with a mouse/pointing device and keyboard);
- minimum RAM: 1 GB or more;
- minimum processor: 1.0 GHz or more;
- the most recent version of Mozilla Firefox, Microsoft Edge, Google Chrome, Safari, Opera (preferred: Google Chrome);
- JavaScript enabled in your browser (mandatory requirement);
- a fast and stable Internet connection (recommended Internet download speed: 1.0 Mbps or higher; recommended Internet upload speed: 0.5 Mbps or higher);
- a color monitor, minimum screen resolution: 640 by 480 pixels (recommended: 1024 by 768 pixels);
- Windows 7/8/10 OS, MacOS X 10.0x or newer, Linux OS;
- whitelist the domains [“*.edube.org”](https://*.edube.org) and [“*.openedg.org”](https://*.openedg.org);
- full access through ports 80 (http), 443 (https), and http redirects permitted.