PCAP: PROGRAMMING ESSENTIALS IN PYTHON (2.0)

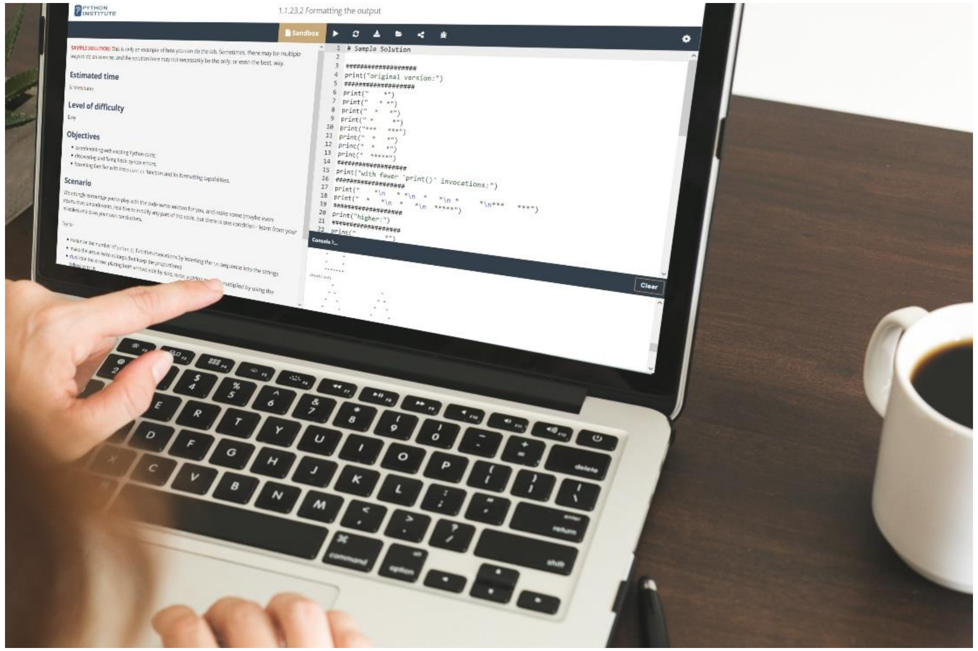
OVERVIEW

DEVELOPED BY

PYTHON INSTITUTE
Open Education & Development Group

CONTENTS

1. OVERVIEW
2. COURSE STRUCTURE
3. SCOPE AND SEQUENCE
4. HOW TO USE THE COURSE
5. CERTIFICATIONS
6. WHY LEARN PROGRAMMING
7. WHY LEARN PYTHON
8. PYTHON EXAMPLES
9. KEY TAKEAWAYS



OVERVIEW

PCAP: PROGRAMMING ESSENTIALS IN PYTHON (2.0)

- ✓ **For beginners** with little or no prior knowledge of programming;
- ✓ Broken down into two parts: **PE1** (Basics) and **PE2** (Intermediate), each part designed to be an independent, one-semester course;
- ✓ 8 x modules, 8 x quizzes, 8 x tests, 30+ lab scenarios, 2 x summary tests, and the final test;
- ✓ **Accessed online** with no special equipment or system requirements;
- ✓ Instructor-led training offered at no cost;
- ✓ Self-paced access offered **at no cost**;
- ✓ Aligned with **PCEP – Certified Entry-Level Python Programmer** (PE1) and **PCAP – Certified Associate in Python Programming** (PE2) certifications.

TARGET AUDIENCE

The *PCAP: Programming Essentials in Python (2.0)* curriculum is designed for students with little or no prior knowledge of programming.

CURRICULUM DESCRIPTION

The *PCAP: Programming Essentials in Python (2.0)* 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.

TARGET CERTIFICATION

The *PCAP: Programming Essentials in Python (2.0)* curriculum helps students prepare for the **PCEP: Certified Associate in Python Programming** (PE1: Modules 1-4) and **PCAP: Python Certified Associate Programmer** (PE2: Modules 1-4) certification exams. *PCEP – Certified Entry-Level Python Programmer* certification is an optional interim step to the *PCAP – Certified Associate in Python Programming* certification.

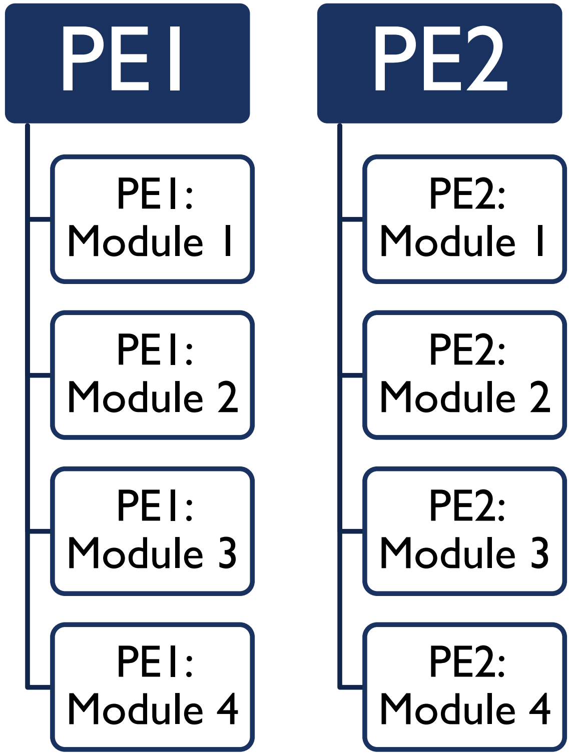
COURSE STRUCTURE

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 – PART 1 (PE1)

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.

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.

PYTHON ESSENTIALS – PART 1 (PE1)

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.

PYTHON ESSENTIALS – PART 1 (PE1)

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.

PYTHON ESSENTIALS – PART 2 (PE2)

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.

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.

PYTHON ESSENTIALS – PART 2 (PE2)

After completing **PE2 Module 3**, the student will:

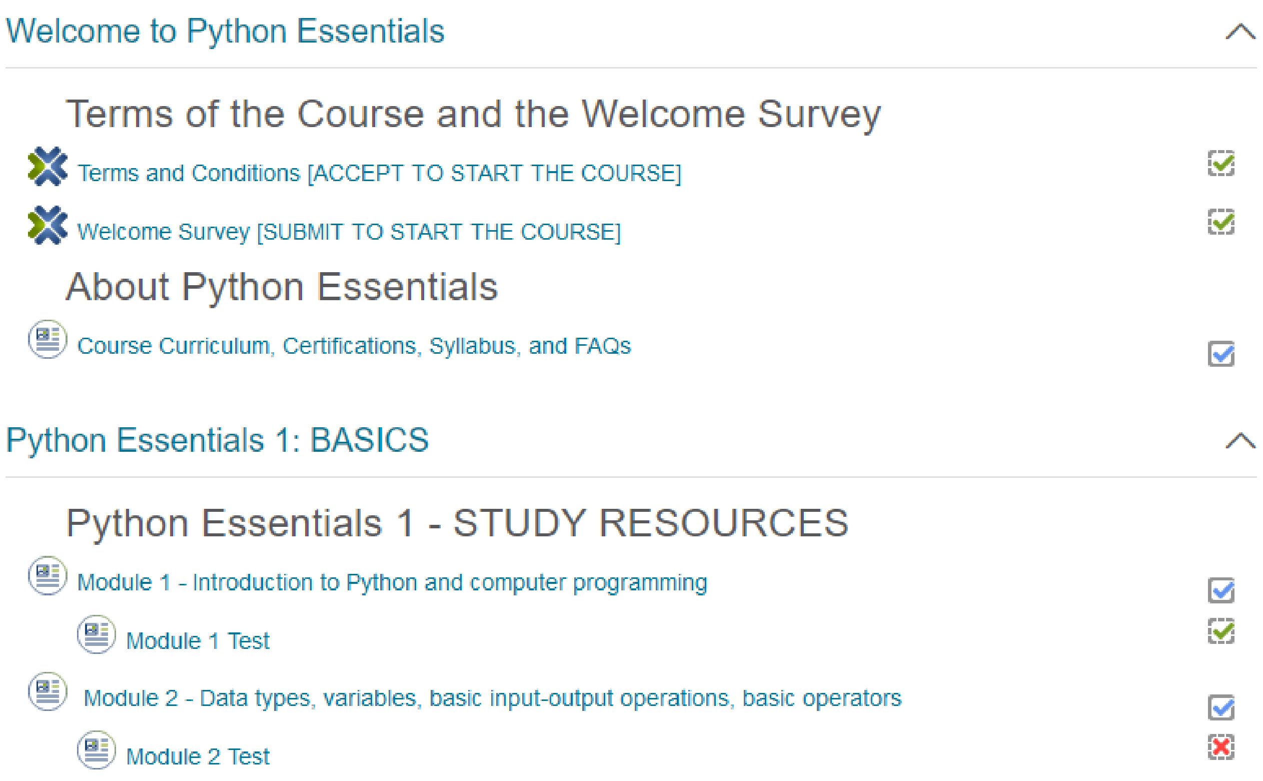
- ✓ understand the fundamental concepts of object programming like class, object, property, method, inheritance and polymorphism;
- ✓ have an orientation in the differences between procedural and object approaches, as well as being oriented when both of the techniques reveal their pros and cons;
- ✓ 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.

PYTHON ESSENTIALS – PART 2 (PE2)

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.

COURSEWARE – NETACAD (MOODLE)



Welcome to Python Essentials

Terms of the Course and the Welcome Survey

Terms and Conditions [ACCEPT TO START THE COURSE] ☑

Welcome Survey [SUBMIT TO START THE COURSE] ☑

About Python Essentials

Course Curriculum, Certifications, Syllabus, and FAQs ☑

Python Essentials 1: BASICS

Python Essentials 1 - STUDY RESOURCES

Module 1 - Introduction to Python and computer programming ☑

Module 1 Test ☑

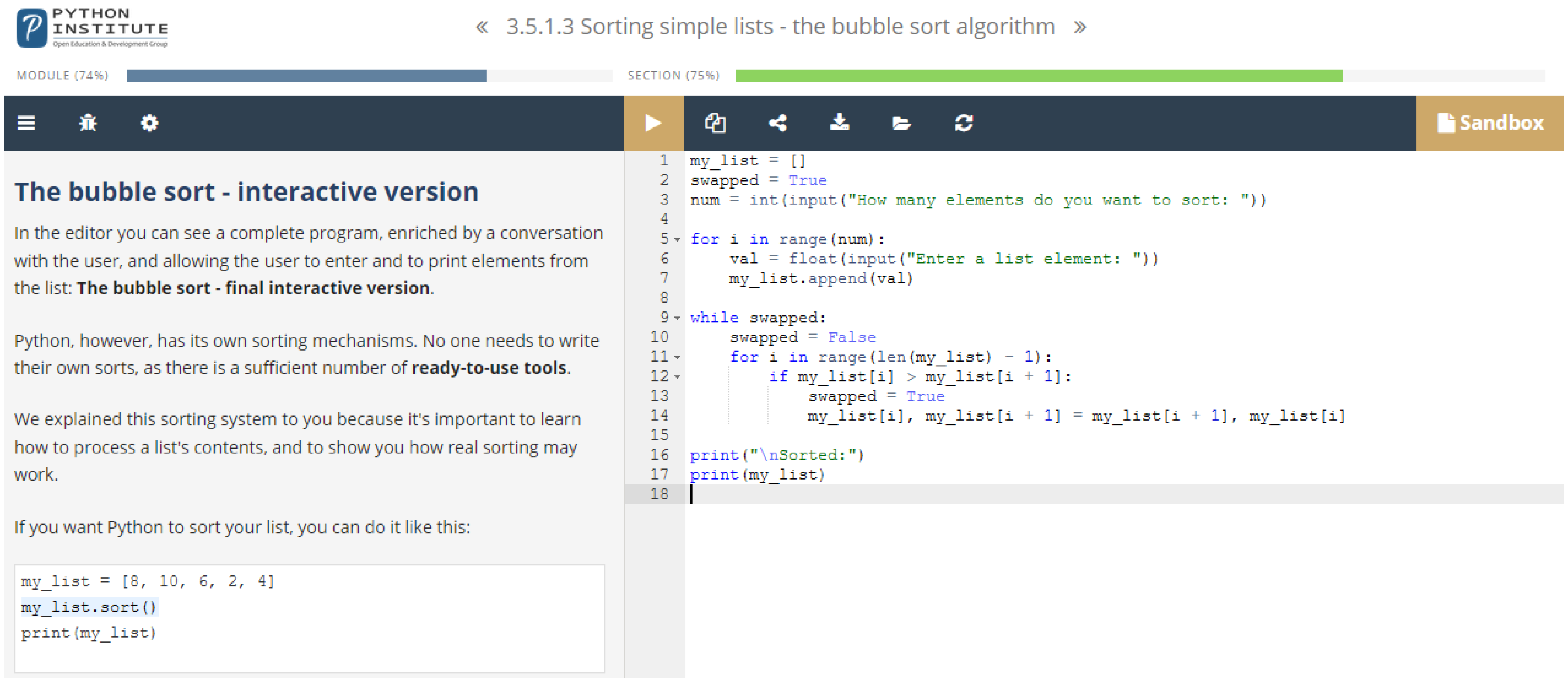
Module 2 - Data types, variables, basic input-output operations, basic operators ☑

Module 2 Test ☒

The main course page on NetAcad:

- ✓ breaks the course down into sections;
- ✓ provides direct links to course resources, study pages and tests;
- ✓ unlocks sections/contents after fulfilling specific conditions (e.g., obtaining a particular test score)
- ✓ Python Essentials – Part 2 is available to those students who complete Python Essentials – Part 1, and pass the Part 1 Summary Test (70% or more).

COURSEWARE – EDUBE INTERACTIVE


PYTHON INSTITUTE
Open Education & Development Group
« 3.5.1.3 Sorting simple lists - the bubble sort algorithm »
MODULE (74%)
SECTION (75%)
Sandbox

The bubble sort - interactive version

In the editor you can see a complete program, enriched by a conversation with the user, and allowing the user to enter and to print elements from the list: **The bubble sort - final interactive version**.

Python, however, has its own sorting mechanisms. No one needs to write their own sorts, as there is a sufficient number of **ready-to-use tools**.

We explained this sorting system to you because it's important to learn how to process a list's contents, and to show you how real sorting may work.

If you want Python to sort your list, you can do it like this:

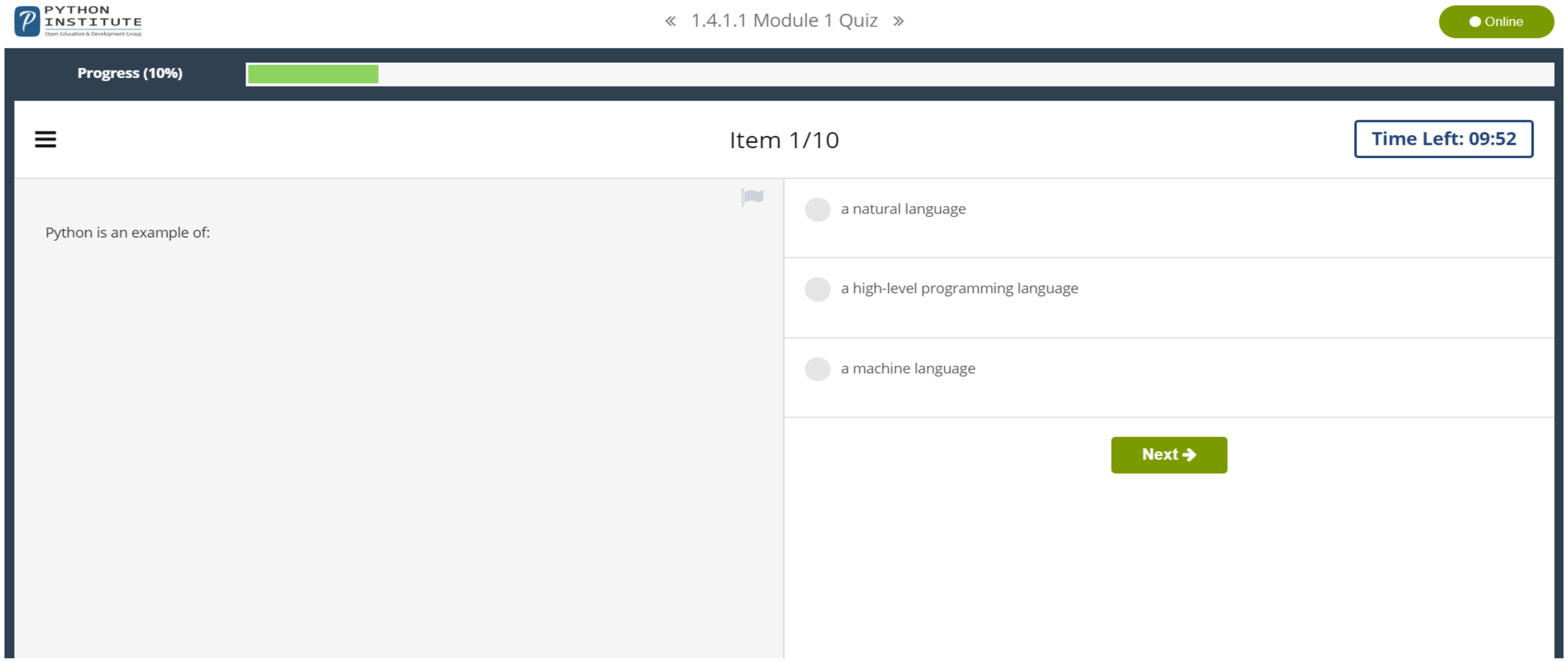
```
my_list = [8, 10, 6, 2, 4]
my_list.sort()
print(my_list)
```

```
my_list = []
swapped = True
num = int(input("How many elements do you want to sort: "))

for i in range(num):
    val = float(input("Enter a list element: "))
    my_list.append(val)

while swapped:
    swapped = False
    for i in range(len(my_list) - 1):
        if my_list[i] > my_list[i + 1]:
            swapped = True
            my_list[i], my_list[i + 1] = my_list[i + 1], my_list[i]

print("\nSorted:")
print(my_list)
```

A resource page:

- ✓ lets the student study and practice code writing as they learn;
- ✓ intuitive and user-friendly interface;
- ✓ programming environment available right from the student’s browser.

COURSEWARE – EDUBE INTERACTIVE



Python is an example of:

○ a natural language

○ a high-level programming language

○ a machine language

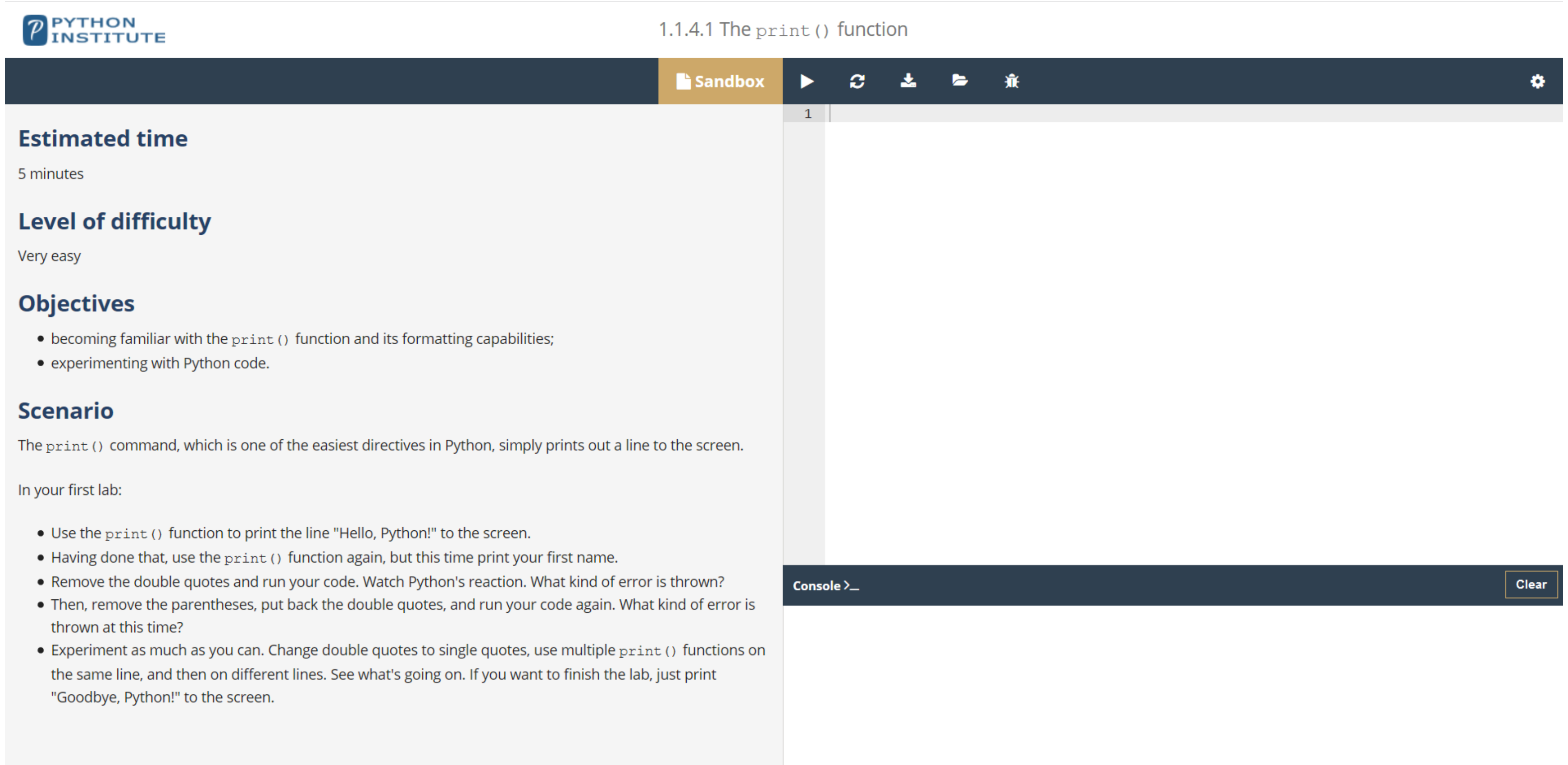
Quizzes and tests:

✓ students can track their progress as they advance through the course;

✓ post-quiz feedback;

✓ test results appear in the gradebook on NetAcad.

COURSEWARE – EDUBE INTERACTIVE



1.1.4.1 The `print()` function

Estimated time

5 minutes

Level of difficulty

Very easy

Objectives

- becoming familiar with the `print()` function and its formatting capabilities;
- experimenting with Python code.

Scenario

The `print()` command, which is one of the easiest directives in Python, simply prints out a line to the screen.

In your first lab:

- Use the `print()` function to print the line "Hello, Python!" to the screen.
- Having done that, use the `print()` function again, but this time print your first name.
- Remove the double quotes and run your code. Watch Python's reaction. What kind of error is thrown?
- Then, remove the parentheses, put back the double quotes, and run your code again. What kind of error is thrown at this time?
- Experiment as much as you can. Change double quotes to single quotes, use multiple `print()` functions on the same line, and then on different lines. See what's going on. If you want to finish the lab, just print "Goodbye, Python!" to the screen.

Lab scenarios:

- ✓ students can practice coding based on real-life programming contexts;
- ✓ mini-programming projects;
- ✓ labs embedded in the course.

SCOPE AND SEQUENCE

CURRICULUM OBJECTIVES

The aim of the course is to:

- spark the student’s interest in computer programming;
- familiarize the student with the universal concepts of computer programming;
- present the Python programming language syntax, semantics, and the runtime environment;
- acquaint the student with general coding techniques and object-oriented programming;
- enable the student to start her/his own studies, and to open a path to the developer’s career.

COURSE SYLLABUS: PYTHON ESSENTIALS – PART 1 (PE1)

PE1: Module 1

Introduction to Python and Computer Programming

✓ 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

PE1: Module 2

Data Types, Variables, Basic Input-Output Operations, Basic Operators

✓ 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

COURSE SYLLABUS: PYTHON ESSENTIALS – PART 1 (PE1)

PE1: Module 3

Boolean Values, Conditional Execution, Loops, Lists and List Processing, Logic and Bitwise Operations

- ✓ 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: constructing, indexing, slicing and content manipulation
- ✓ How to sort a list using a bubble-sort algorithm
- ✓ Multidimensional lists and their applications

PE1: Module 4

Functions, Tuples, Dictionaries, and Data Processing

- ✓ 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

PE1 GRADUATE PROFILE

A student who has completed **Python Essentials – Part 1** (PE1) 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.

COURSE SYLLABUS: PYTHON ESSENTIALS – PART 2 (PE2)

PE2: Module 1

Modules, Packages, and PIP

- ✓ 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

PE2: Module 2

Exceptions, Strings, String and List Methods

- ✓ 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

COURSE SYLLABUS: PYTHON ESSENTIALS – PART 2 (PE2)

PE2: Module 3

Object-Oriented Programming

- ✓ 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

PE2: Module 4

Miscellaneous

- ✓ Generators, iterators and closures
- ✓ Working with filesystem, directory trees and files
- ✓ Selected Python Standard Library modules (*os*, *date*, *datetime*, *calendar*)

PE2 GRADUATE PROFILE

A student who has completed **Python Essentials – Part 2** (PE2) 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.

HOW TO USE THIS COURSE – ACADEMIC INSTITUTIONS

Academic institutions can use this course as follows:

- offer the course as a complete two-semester course (or offer PE1/PE2 only as a one-semester course)
- create interest and motivate new students to learn the fundamentals of computer programming;
- motivate those students who already know another programming language to learn Python;
- supplement an existing Python language course;
- help students prepare for the *PCEP – Certified Entry-Level Python Programmer* and *PCAP – Certified Associate in Python Programming* certifications*;*
- introduce NetAcad and Edube Interactive to peers and colleagues.

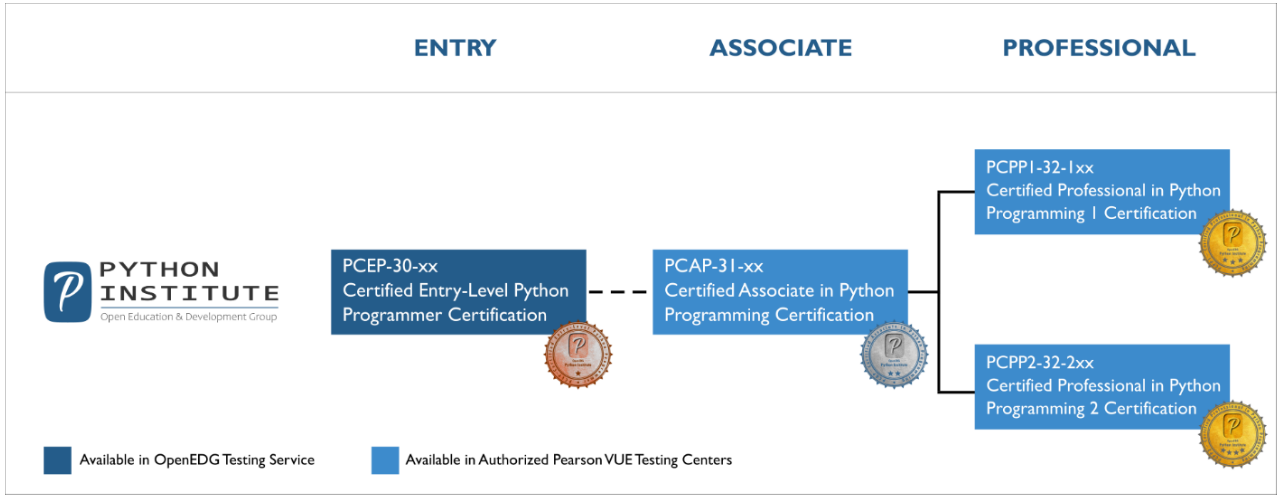
There are **no formal requirements for instructors** to teach *PCAP: Programming Essentials in Python (2.0)*. However, the Python Institute recommends that instructors earn a *PCAP – Certified Associate in Python Programming* certification prior to teaching the class.

CERTIFICATIONS

OpenEDG Python Institute has defined an independent global certification path for the Python programming language, and has developed an international Python programming examination standard.

The entry-level certification exams are administered worldwide in the form of proctored (OpenEDG Authorized Testing Centers) and non-proctored (accessed online from any place) tests through the OpenEDG Testing Service program, at a time and location convenient to test candidates.

The associate-level certification exams are administered in the form of proctored tests at over 5,000 Pearson VUE® Authorized Test Centers, and online via OnVUE Online Proctoring from Pearson VUE®, at a time and location convenient to test candidates.



ENTRY

ASSOCIATE

PROFESSIONAL

PYTHON INSTITUTE
Open Education & Development Group

PCEP-30-xx
Certified Entry-Level Python Programmer Certification

PCAP-31-xx
Certified Associate in Python Programming Certification

PCPP1-32-1xx
Certified Professional in Python Programming 1 Certification

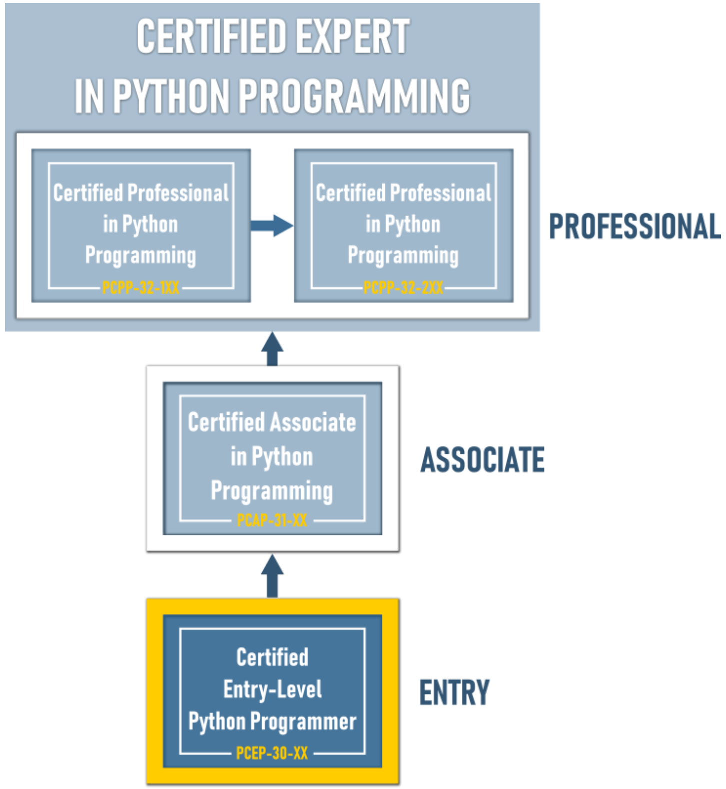
PCPP2-32-2xx
Certified Professional in Python Programming 2 Certification

■ Available in OpenEDG Testing Service

■ Available in Authorized Pearson VUE Testing Centers

https://pythoninstitute.org/certification

CERTIFICATIONS



CERTIFIED EXPERT IN PYTHON PROGRAMMING

Certified Professional in Python Programming PCPP-32-1XX

Certified Professional in Python Programming PCPP-32-2XX

PROFESSIONAL

Certified Associate in Python Programming PCAP-31-XX

ASSOCIATE

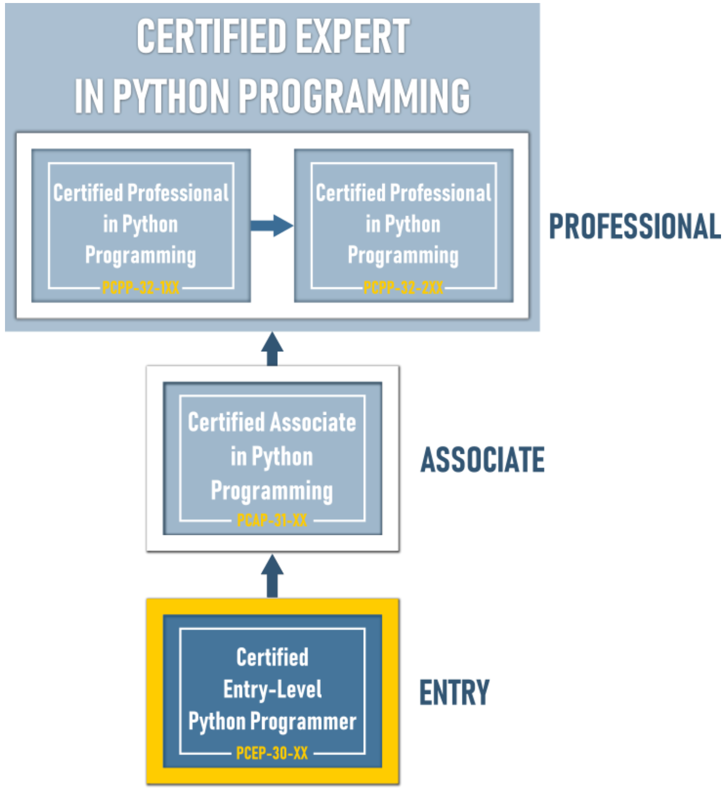
Certified Entry-Level Python Programmer PCEP-30-XX

ENTRY

PCEP – Certified Entry-Level Python Programmer certification shows that the individual is familiar with universal computer programming concepts like data types, containers, functions, conditions, loops, as well as Python programming language syntax, semantics, and the runtime environment.

- ✓ Professional certification
- ✓ Entry level
- ✓ Delivered via OpenEDG Testing Service (edube.org)
- ✓ Digital certification issued by OpenEDG Python Institute
- ✓ Digital badge issued on Credly's Acclaim
- ✓ Complete PE1: Modules 1-4 from the *PCAP: Programming Essentials in Python* course and prepare for the exam

CERTIFICATIONS



CERTIFIED EXPERT IN PYTHON PROGRAMMING

Certified Professional in Python Programming PCPP-32-1XX

Certified Professional in Python Programming PCPP-32-2XX

PROFESSIONAL

Certified Associate in Python Programming PCAP-31-XX

ASSOCIATE

Certified Entry-Level Python Programmer PCEP-30-XX

ENTRY

PCAP – Certified Associate in Python Programming certification is a professional credential that measures the 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.

- ✓ Professional certification
- ✓ Associate level
- ✓ Delivered in partnership with Pearson VUE
- ✓ Digital certification issued by OpenEDG Python Institute
- ✓ Digital badge issued on Credly’s Acclaim
- ✓ Complete PE2: Modules 1-4 from the *PCAP: Programming Essentials in Python (2.0)* course, prepare for the exam, and get a **50% discount code**

WHY LEARN PROGRAMMING

FOR SEVERAL REASONS:

- ✓ To become a **creator**: a highly **creative** and **powerful** one. Go as far as your imagination lets you.
- ✓ Strong programming skills are a **hot commodity** on the job market!
- ✓ Boost your **earning potential**!
- ✓ Programming is the language of the **future**.
- ✓ Learning to program means learning to **think in abstract** and more **precise ways**.
- ✓ It helps you **do better in other areas**!
- ✓ It's **fun**!

WHY LEARN PYTHON

FOR A MILLION REASONS! Here are just some of them:

- It is **omnipresent** – people use numerous Python-powered devices on a daily basis, whether they realize it or not.
- There have been millions (well, actually **billions**) of lines of code written in Python, which means almost unlimited opportunities for **code reuse** and learning from well-crafted examples.
- It is **easy to learn** – the time needed to learn Python is shorter than for many other languages; this means that it's possible to start the actual programming faster.
- It is **easy to use** for writing new software – it's often possible to write code faster when using Python**.**
- It is **easy to obtain**, install and deploy – Python is free, open and multiplatform; not all languages can boast that.
- There is a **large** and very **active Python community.**
- It gives you a **solid foundation** and allows you to learn other programming languages (e.g., C++, Java, or C) much easier and much faster.
- It's **fun**!

KEY TAKEAWAYS

- *PCAP: Programming Essentials in Python 2.0* (short: *Python Essentials*) is developed by OpenEDG Python Institute;
- The course introduces students to computer programming using the Python language;
- The course aligns with the *PCEP – Certified Entry-Level Python Programmer* and *PCAP – Python Certified Associate Programmer* certifications;
- Test candidates who pass PCEP and/or PCAP exam(s) receive a digital certification, score report, and a digital badge;
- The Python Institute provides all contents;
- The course is available in NetAcad;
- Students who successfully complete the course and pass the final test will receive a 50% discount for the *PCAP – Certified Associate in Python Programming* certification exam at Pearson VUE/OnVUE Online Proctoring.
- Visit www.pythonsinstitute.org for more information about the certification program.