



## **PROGRAMMING ESSENTIALS IN PYTHON | PCAP Certification**

**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 will prepare the student for jobs/careers connected with widely understood software development, which includes not only creating the code itself as a junior developer, but also computer system design and software testing.

It could be a stepping-stone to learning any other programming language, and to explore technologies using Python as a foundation (e.g., Django, SciPy).

This course is distinguished by its affordability, friendliness, and openness to the student. It starts from the absolute basics, guiding the student step by step to complex problems, making her/him a responsible software creator able to take on different challenges in many positions in the IT industry.

### **TARGET AUDIENCE**

Programming Essentials in Python curriculum is designed for students with little or no prior knowledge of programming.

### **TARGET CERTIFICATION**

Programming Essentials in Python curriculum helps students prepare for the PCAP | Python Certified Associate Programmer certification exam. PCAP 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.

### **PCAP – COURSE MODULES & OBJECTIVES**

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 relations and bindings.
- 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 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 content 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.

## 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 will help you do better in other areas!

## WHY LEARN PYTHON?

- 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 will give you a solid foundation and allow you to learn other programming languages (e.g., C++, Java, or C) much easier and much faster.
- It will be fun!

## PYTHON IS A GREAT CHOICE FOR

- Web and Internet development (e.g., Django and Pyramid frameworks, Flask and Bottle micro-frameworks)
- Scientific and numeric computing (e.g., SciPy – a collection of packages for the purposes of mathematics, science, and engineering; Ipython – an interactive shell that features editing and recording of work sessions)
- Education (it's a brilliant language for teaching programming!)
- Desktop GUIs (e.g., wxWidgets, Kivy, Qt)
- Software Development (build control, management, and testing – Scons, Buildbot, Apache Gump, Roundup, Trac)
- Business applications (ERP and e-commerce systems – Odoo, Tryton)
- Networking Automation (NETCONF, RESTCONF, APIC-EM, APIC)

## **SUMMARY**

- PCAP | Programming Essentials in Python is developed by the Python Institute;
- The course introduces students to computer programming using the Python language;
- The course aligns with the Python Institute PCAP | Python Certified Associate Programmer certification;
- The Python Institute provides all course content;
- Students who successfully complete the course work and pass the final test may request to receive a 50% discount for the PCAP | Python Certified Associate Programmer certification exam at Pearson VUE. (n.b. exam discount is subject to change)

## **CLASS DURATION**

5 Days – Monday to Friday (8.30am – 4.30pm)

## **INTERESTED?**

Send an inquiry to: [training@afraiti.org](mailto:training@afraiti.org)

Or visit website: <http://www.afraiti.org/python/>