

# **Outline:** Introduction to Python 3 for IC Design Engineers

# Course Description:

This <u>three-day</u> course is a practical introduction to Python 3 for design, evaluation, testing or production support engineers. This course is intended to give attendees a clear grasp of the fundamentals of Python programming and the skills to use that knowledge to automate tasks and derive insights from data.

Using these fundamentals, participants will also be guided on how to use an AI Assistant to improve efficiency when writing code and to add AI features to a Python application.

### Intended Audience:

Design, evaluation, testing or production support engineers with some programming experience (in another language) will benefit. Participants without a formal programming background can also gain.

## **Learning Outcomes:**

At the conclusion of this course, attendees will be able to:

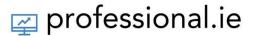
- Design and programme Python applications using Visual Studio Code and Jupyter environments
- Use the main flow of control elements in Python
- Choose the appropriate variable type when required
- Use the different collection types, including lists, tuples and dictionaries
- Write functions and pass parameters
- Create classes and objects
- Read, write and parse different types of files
- Access operating system variables and automate tasks
- Use Numpy and Pandas to represent data sets
- Graph using Matplotlib and other tools
- Read and write to a SQLite database
- Create a basic UI using TKInter
- Use an AI Assistant to improve efficiency when writing code
- Take first steps to adding AI to your Python application using the ChatGPT API

#### Course materials:

All participants will receive course example files, cheat-sheets and PDF reference materials prior to the course or on the first morning. Specifically, they will receive:

- Python Programming e-Book
- GitHub repository with all course code and other materials
- Exercises with Solutions for all topics
- Cheat sheets for handy reference
- Python language Reference Documentation
- Separate Case Studies for Self-Study
- Extra references and resources for material not covered in the provided text

www.professional.ie 1



## **Software Environment:**

Python with Visual Studio Code with Python and Jupyter Notebooks extensions. Pycharm, Spyder and other Python development tools and environments like Anaconda can be used if required by participants.

# **Labs and Exercises**

This is a practical hands-on course where participants write, test and debug code throughout. Exercises and examples are tailored to the practical needs and backgrounds of electronic engineers. Lab exercises are completed by participants and overall about 35% of the course time is used for hands-on exercises and 65% for demonstration.



# **COURSE CONTENT:**

# Day 1 – Introduction to Python 3

## **Python Basics**

The Python environment

Visual Studio Code Integrated Development Environment

Variables

Keywords

Variable types

**Built in functions** 

The Zen of Python and writing "Pythonic" code

#### Flow Control

if and elif

Conditional expressions

Relational operators

**Boolean operators** 

while loops

for loops

Using break and continue

for else and while else loops

#### **Functions**

Defining a function

**Function parameters** 

Returning values

Default parameter values

Named and positional parameters

Global variables

Variable scope

# **Modules and Packages**

Python standard library

The import statement

Creating your own module

Understand \_\_name\_\_ and how to write test code for your module

Third party library modules

The Python package manager - pip

www.professional.ie

3



# Lists, Tuples

About sequences
Lists
Indexing and slicing
Iterating through a sequence
Functions for all sequences
Using enumerate
Operators and keywords for sequences
List comprehension techniques
Generators

# **Working With Strings**

Creating strings
String functions
fStrings
Indexing and slicing strings
t-strings (new in Python 3.14)
Understanding immutable and mutable objects
The id() function

# **Using AI Assistants for Coding**

Overview of AI Warnings Prompt engineering techniques Getting ChatGPT or Copilot to write code Iterative development



# Day 2 - Introduction to Python 3

# Working with files

Text file I/O
Opening a text file
The with block
Reading a text file
Writing to a text file

## **Exception Handling**

Exceptions
Handling exceptions with try
Handling multiple exceptions
Handling generic exceptions
Ignoring exceptions
Using else
Cleaning up with finally
re-raising exceptions
Raising a new exception
The standard exception hierarchy

### **Dictionaries and Sets**

About dictionaries & When to use dictionaries Creating dictionaries Getting dictionary values Iterating through a dictionary Reading file data into a dictionary

# **OS Services and Task Automation**

The sys module & The os module
Using argy to access command line parameters
Environment variables
Launching external processes
Paths, directories, and filenames

#### **SQLite Introduction**

Relational database overview
SQLite module
Connecting to a database
Querying data
Insert, update and delete operations

## **Basic UI Applications**

Creating a simple desktop UI application The TKInter module A basic window Adding Widgets Event driven programming



6

# Day 3 - Introduction to Python 3

#### Classes

Defining classes
Instance objects & Instance attributes
Methods & Properties
Class data
Inheritance
Pseudo-private variables
Static methods

# Adding AI To Your Python Application Using ChatGPT API

Making an API call from Python Introducing the ChatGPT API Making a request Simple example - sentiment analysis

### Juypter

The Visual Studio Code Jupyter extension Understanding markdown Tab completion Magic commands Benchmarking External commands & Enhanced help Notebooks

## NumPy

Objectives
Python's scientific stack
NumPy overview
Creating arrays & Creating ranges
Working with arrays
Shapes
Slicing and indexing
Indexing with Booleans

#### **Pandas**

About Pandas
Architecture & Series
DataFrames
Index Objects
Basic Indexing
Broadcasting

## Matplotlib

Setting up Matplotlib Creating different plot types Customizing your plot. Alternatives to Matplotlib – e.g Seaborn

www.professional.ie



7

# Day 4 - Certification preparation day following Python 3 Introduction course

# Title: Certified Entry-level Python Programmer Exam Preparation

Duration: 1 Day Delivery

# **Administration & Support**

Between the completion of the three-day course and the PCEP preparation day, all participants will be emailed weekly to offer support and encouragement to complete the six Udemy practice exams provided. They will also be registered for the PCEP Practice exams and Official Certification exam and offered all assistance to ensure their success.

# **Target Audience:**

This course is intended to prepare attendees for the Python Institute PCEP Exam, PCEP-30-02.

Details on the exam can be found here:

PCEP - Certified Entry-Level Python Programmer Exam - PCEP-30-02

Attendees should have completed the Introduction to Python course.

## **Learning Outcomes:**

At the conclusion of this certification preparation day, attendees will be ready to sit the PCEP Exam.

# Content for Delivery Day:

Section 1: Revision of Key Concepts with sample questions (Morning Session)

- Fundamental Terms
- Pythons Logic and structure
- Variable types
- Operators
- Input/Output
- Flow of control
- Lists
- Tuples
- Dictionaries
- Strings
- Functions
- Exceptions

## Section 2: Practice Exam (Early Afternoon)

# Section 3: Attendees sit PCEP-30-02 certification exam (Afternoon)

www.professional.ie