Table of Contents

[1 BASIC COURSE CONTENTS 2](#_Toc526318998)

[2 INTERMEDIATE COURSE CONTENTS 4](#_Toc526318999)

[3 INDUSTRY SPECIFIC PYTHON TOPICS (Advanced) 6](#_Toc526319000)

[3.1 Database Management 6](#_Toc526319001)

[3.2 Python and Data Sciences with NumPY and PANDAS – An Intro 6](#_Toc526319002)

# BASIC COURSE CONTENTS

## Overview

###### **Python is a programming language that has been around since the 1990s.**

From the very beginning, Python has been designed to be compact, modular, efficient and extensible.

Additional functionality can be easily incorporated through the large standard library that Python itself provides and third party modules/libraries developed by the community.

This ability to constantly enhance/extend the functionality of Python has made it very popular with the Developer community.

An entire ecosystem has grown up around Python and it is being increasingly used for enterprise web application development, machine learning, data sciences and other kinds of applications. More and more enterprises are looking at Python to solve their business problems.

This course requires 35 hours and includes approximately 40 examples, 20 exercises that the students will have to complete on their own and 2-4 activities that they will have to execute in groups to foster the concept of collaboration.

Using a code based interactive approach, as opposed to theory first, a more effective learning environment will be provided. From day one, we will dive straight into code !.

We are confident about making the students excited about this great language, curious about exploring features on their own and becoming part of the vibrant Python community !.

###### **Next Steps** – After completing this course, the students can immediately move onto more advanced topics

## Course Contents

|  |  |  |  |
| --- | --- | --- | --- |
| **General Introduction** | | | |
|  | **Informal introduction to Python** |  |  |
|  | **Installation instructions** |  |  |
|  | **Python Ecosystem** |  |  |
|  | **The interactive shell** |  |  |
|  | **Widely used editors (IDLE, etc.)** |  |  |
|  | **Some sample programs** |  |  |
|  | **Reading material that students can refer to for further study** |  |  |
| **Managing your data (Data is everywhere. Much of programming is all about handling data)** | | | |
|  | **Strings** |  |  |
|  | **Numbers** |  |  |
|  | **Integers** |  |  |
|  | **Boolean Values** |  |  |
|  | **Data structures** |  |  |
|  | **Custom data types** |  |  |
| **Conditional Execution (Deciding When to Run Blocks of Code)** | | | |
|  | **If statements** |  |  |
|  | **If-Elif** |  |  |
|  | **If-Else** |  |  |
| **Iteration (Processing data sequentially, Looping through data sets)** | | | |
|  | **While loops** |  |  |
|  | **For Loops** |  |  |
|  | **Use file processing as an example** |  |  |
| **Functions in Python (Avoiding duplication of code, creating re-usable modular code)** | | | |
|  | **Some sample functions** |  |  |
|  | **Importance of functions** |  |  |
|  | **Arguments to a function** |  |  |
|  | **\*args,\*\*kwargs** |  |  |
|  | **Lambda Introduction** |  |  |
|  | **Function decorators** |  |  |
| **A deep dive into Data Structures** | | | |
|  | **Lists** |  |  |
|  | **Tuples** |  |  |
|  | **Dictionary** |  |  |
|  | **Sets** |  |  |
|  | **Using lists as Stacks, Queues, etc.** |  |  |
|  | **Looping through data structures** |  |  |
|  | **A brief intro to some more advanced data structures** |  |  |
| **Exception Handling and Logging (No matter how good your code is,things do go wrong at run time.How to handle these situations )** | | | |
|  | **Importance of writing good code with logging and exception handling** |  |  |
|  | **Raising exceptions** |  |  |
|  | **User defined exceptions** |  |  |
|  | **Handling exceptions** |  |  |
|  | **Logging information for debugging purposes** |  |  |
| **Python Coding Guidelines (Coding standards and static code analysis are mandatory in most LIVE projects | Introduction to PyLint – static code checker)** | | | |
|  | **PEP8** |  |  |
|  | **Static Code Analysis** |  |  |
|  | **Python modules for actually doing static analysis** |  |  |
| **Modules (Creating small compact programs, logically grouping functionality, sharing of code)** | | | |
|  | **What is a module?** |  |  |
|  | **Creating modules** |  |  |
|  | **Third party modules** |  |  |
|  | **Installing third party modules** |  |  |
|  | **Using multiple modules in your code** |  |  |
|  | **Design considerations** |  |  |
| **Packages (Logically structuring your application code using Folders)** | | | |
|  | **Design considerations** |  |  |
|  | **Introduction to packages** |  |  |
|  | **Different ways to group applications/modules into packages** |  |  |
|  | **\_\_init\_\_.py** |  |  |
| **Classes (Object oriented features provided by Python)** | | | |
|  | **Introduction to OOP** |  |  |
|  | **Polymorphism** |  |  |
|  | **Inheritance** |  |  |
|  | **Constructors** |  |  |
|  | **Objects** |  |  |
|  | **Getter/Setters** |  |  |

# INTERMEDIATE COURSE CONTENTS

| **#** | **TOPIC** | **SUB TOPICS** | **LEVEL** | **HRS** | **REMARKS** |
| --- | --- | --- | --- | --- | --- |
| 1 | Python’s standard library | * Walkthrough of the standard library * Using Python standard library for file processing * String operations * JSON processing * Mathematical operations | Intermediate | 8 | * Even the simplest python module will use the standard library * This will provide an introduction which developers can build upon |
| 2 | Advanced Data Structures/ Collections | * Review data structures/collections covered in the basic course * Third party modules for using various types of collections/data structures * Multi dimensional data structures | Intermediate | 12 | * Understanding of advanced data structures is vital for using Python in Data sciences * Pandas and NumPy are two widely used third party modules * These are deep concepts that require detailed explanations |
| 3 | Virtual Environments | * What are virtual environments * Setting up a virtual environment * Using multiple virtual environments * Setting up projects with packages * Using pip to install modules in virtual environments | Intermediate | 8 | * In practice, Python developers will develop code using virtual environments * Developers must be able to setup and use virtual environments |

# INDUSTRY SPECIFIC PYTHON TOPICS (Advanced)

## Database Management

| **#** | **TOPIC** | **SUB TOPICS** | **LEVEL** | **HRS** | **REMARKS** |
| --- | --- | --- | --- | --- | --- |
| 1 | Using Python with PostgreSQL | * Installing and using third party modules * Performing various database operations |  | 12 | * PostgreSQL is a widely used relational database |
| 2 | Using Python with MongoDB | * Installing and using third party modules * Performing various database operations |  | 12 | * MongoDB is a NoSQL database |

## Python and Data Sciences with NumPY and PANDAS – An Intro

| **#** | **TOPIC** | **SUB TOPICS** | **LEVEL** | **HRS** | **REMARKS** |
| --- | --- | --- | --- | --- | --- |
| 1 | Using Python with NumPY and PANDAS | * Installing and using the relevant third party modules * Introduction to Pandas and NumPY * High performance data structures available * Series, data frames, Panels and other concepts |  | 16 |  |