





Python for Development

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **Approved By** | **Remarks/Revision Details** |
| 1 | 22/05/2023 |  |  |  | Initial Release |
| 2 | 13/06/2023 |  |  |  | Added TDD in the outline  Included Optional Day 6 Content |
| 3 | 20/03/2024 | Ramakant Debata |  |  | Updated Indicative hours and Course Resources as per discussion |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Document History**

**Course Title:**

# Course Summary:

The course aims at explaining the practical usage of python for development. The course discusses the fundamental syntax and philosophy of python along with its core programming elements.

1. **Pre-Requisite**

Basic understanding of programming would be preferrable.

1. **Audience**

Engineers who are looking forward to start programming.

1. **Hardware & Network Requirements**

* Any quad core CPU or above
* Windows or mac
* 8gb RAM or above
* Basic internet connection

1. **Software Requirements**

* Python 3
* VS Code

# Learning Outcomes:

* Understand the basic language structure of python
* Python fundamental system
* Python collection
* Python object-oriented programming

# Course Content (day wise):

## Day 1

### Introduction

* What’s Python?
* Why do people use Python?
* Python Ecosystem
* Python Versioning
* Installing
* Switching
* Python IDE

### Hello World Python

* Python Shell (REPL)
* Writing simple scripts
* Python 2 vs 3 Differences
* Executing Python Scripts

### Python – Getting Started

* Python Statements
* Variables
* Operators and Expressions
* Datatype
* Object and Id
* Mutable and Immutable

### Python Functions

* Function Basics
* Defining Functions
* Calling Functions
* Scopes

### Python's Lists

* Common List Methods
* The range() Function
* List Operations
* String Indexing
* String Slicing
* String Iteration
* Multi-Dimensional Lists (Matrices)

### Python String Types

* Generating Strings in Python
* Immutable
* Common String Methods
* Type Conversion in Python
* Formatting String Output
* Format Specifier
* Variable Substitution
* String Indexing
* String Slicing
* String Iteration

### Python's Tuples

* Immutable
* Common Tuples Methods
* Tuples Operations
* Tuples Indexing
* Tuples Slicing
* Tuples Iteration
* Multi-Dimensional Tuples (Matrices)

### Python Dictionaries

* Python Dictionaries
* Assigning Values to Dictionaries
* Dictionary Methods
* Dictionaries vs Lists & Tuples
* Dictionary Indexing
* Dictionary Iteration

## Day 2

### More on Python Functions

* Function Basics
* Defining Functions
* Argument Defaults
* Lambdas
* Local Variables
* Understanding \_\_builtin\_\_
* Preventing Variable Modifications
* Variable Args
* Keyword Argument Methods

### Object Oriented Programming

* Introduction to OOP using python
* Classes and class attributes
* Instances and instance attributes
* Initialization and cleanup
* Binding and method invocation
* scopes
* Composition and Subclasses
* Built-in functions for classes, instances and other objects

## Day 3

### Exceptions

* About Exceptions
* Python's Default Exception Handler
* Using Try/Except/Else/Finally Exceptions
* Generating User Defined Exceptions
* More on Exceptions
* Exception Examples

### Modules & Packages

* Module Basics
* Packages
* Using \_\_all\_\_ and \_ Variables
* Using \_\_name\_\_
* Using third party modules

### Standard Python modules

* Using the sys module
  + sys.argv, sys.path, sys.version
* An overview on \_\_builtin\_\_ and \_\_future\_\_ modules
* Using the os module
* Filesystem/directory functions

## **Day 4**

### Basic Input/Output with Files

* Opening Files
* Working with Files
* Controlling Output Location

### XML and JSON

* Working with XML
* DOM and Sax
* Introducing ElementTree
* Parsing XML
* Navigating the document
* Creating a new XML document
* JSON
* Parsing JSON into Python
* Converting Python into JSON

## Day 5

### Regular expressions Overview

* Introduction to regexps
* Special symbols and characters for RE
* Metacharacters and Metasymbols
* Practical examples

### Unit Testing and TDD

* What is Unit Testing
* Why is it important
* Unit Testing Framework
* Unit Testing anatomy
* TDD
* Red-Green-Refactor

# Course Structure:

|  |  |
| --- | --- |
| **Activity** | **Indicative Number of Hours** |
| Pre-Read Hours | n/a |
| Teaching Hours | 32.5 |
| Hands on Sessions Hours | 02.5 |
| Assignments & Tutorial Hours | n/a |
| Mock Project Hours | n/a |

# Course Structure:

|  |  |  |
| --- | --- | --- |
| **Method of Assessment** | **Yes/No** | **Weightage** |
|  |  |  |
| Pre-Assessment | Yes |  |
| Mid-Assessment | No |  |
| Post-Assessment | Yes | 100% |
| Project Work | n/a |  |

# Course Resources:

* 1. **Code Samples: Sample code snippets and solutions for better understanding.**
  2. **Assignments: Practical assignments to reinforce learning and build real-world skills.**

# Recommended Reading Links:

# Course Owner (s):

|  |  |  |
| --- | --- | --- |
| **Employee Name** | **Employee Mail ID** | **Business Unit** |
|  |  |  |
|  |  |  |
|  |  |  |