DTC – Programming in Python

Sifal, Kathmandu Phone No.: 01-446-7153

Programming in Python: LEVEL 1

OBJECTIVES

The DTC – Programming in Python- Level 1 course is targeted for beginners who want to:

- Learn how to think and write meaningful piece of code in Python.
- Learn how to read Python code that has been written by somebody else.
- Learn how to map literary description of a problem (requirement) to an application/library coded in Python.

In summary, this course teaches how to program using Python programming language. This is a core basic level course that is essential for anyone who have no prior programming experience but wish to be a professional Python engineer in future

TARGET GROUP

- Anyone who has some basic knowledge about programming and wants to learn to write applications in PYTHON for any purpose e.g. curiosity, hobby, to complete an academic project, to work towards a career as Python programmer, to help in project management, etc.

Prerequisites:

- Basic knowledge about programming, bits/bytes, procedures, classes, computer architecture, etc. If you just have a theoretical knowledge that is perfectly okay but you should have strong convictions on what programming is, and what you hope to achieve from this class.
- Willing and eager to spend at least 10-20 hours (varying from student-to-student) per week outside of the training class to read/write codes in Python (self-study and practice).
- There is no prior educational level requirement for this course. Anyone from 10+2 student to someone who is doing her PHD in Genetic Engineering is welcome to take this course.
- If you are only interested in theory and have no interest/patience in spending at least 10 hours every week throughout the duration of the course, then this course is clearly not for you.
- If you have absolutely no idea about programming or do not see yourself doing programming in the next six -odd months, then this class may not be for you!

TRAINING METHOD

- The course is spread over 40 hours that consists of lecture and lab work. There will be approximately 10 hours of lectures and 30 hours of hands-on lab work.
- Lab exercises are mandatory, have a fixed deadline, and are graded. The course puts heavy emphasis on lab exercises because software programming can only be learnt well by explicitly putting into practice the principles that have been taught (i.e. in simpler terms – by doing lots and lots of coding). Late submission (past the deadline) of exercises incur some penalty from total points.
- Instructors may provide relevant lecture/lab notes to students as (and when) necessary in the form of printed handouts and or via emails.
- Instructors may provide supplementary code snippets to students via email or in lab class to support the theory and or lab material that is being taught.
- At the end of the course, students may have to give an exam (which will be optional), that will test their knowledge on the material covered during the course. This exam may be practical and/or theoretical and is mandatory for any student wishing to join a higher level.
- Students are graded on the basis of attendance, lab exercises and exam in the increasing order of importance.

In summary, the only effective way to learn programming is to write lots of code. So, in order to really make this training productive, students are encouraged to spend as much time as necessary to complete the lab exercises on time. As part of the course, students will spend at least 30 hours in the lab but especially if you are new to programming or are coming from a non-computer-science background, it is recommended that you spend at least 10-20 hours per week outside of the class on your own to practice coding in Python.

COURSE DURATION

- 30 hours
- Classes
  - ✓ Morning/Evening

COURSE BREAKDOWN

Theory:

1. OVERVIEW OF PYTHON LANGUAGE
   - Introduction
   - H/w and s/w requirements
   - Installation of python
   - Using interpreter

2. CORE DATA STRUCTURES
   - String, variables
   - Tuples
   - List
   - Dictionary
   - Operation on data structures
   - Slicing

3. CONSTANT, VARIABLES AND DATA TYPES
   - Primitives and non-primitives variables

4. DECISION AND BRANCHING
   - IF, ELSE, SWITCH, BREAK, CONTINUE

5. LOOPING
   - FOR, WHILE, DO-WHILE

6. FUNCTIONS
   - Building modules
   - Functions
   - Function types
   - Lambdas
   - Map/filter
   - Comprehension
   - For, while, do-while

7. EXCEPTION HANDLING
   - Introduction
   - Handling exceptions
   - Raising exception
   - Catching exceptions
   - Chaining exceptions

8. OOPS
   - Introduction to class/objects
   - Writing a class
   - Inheritance
   - Polymorphism

- Encapsulation
- Operator overloading
- Working with database

LABS

Lab assignments will focus on the practice and mastery of contents covered in the lectures; and introduce critical and fundamental problem-solving techniques to the students.

DISCLAIMER

Please note that Deerwalk Training Center reserves the right to change the course syllabus of DTC – Programming in Python – Level 1 course at any time without prior notification.

Programming in Python: LEVEL 2

OBJECTIVES

The DTC – Programming in Python – Level 2 course is targeted for trainees:

- Who have had some prior beginner level hands-on programming experience in Python programming language.
- Who have programming experience in some other programming language (e.g. Python, Obj-C, PHP, C, C++, etc.) and want to learn Python.

TARGET GROUP

- High school and university students (undergraduate, graduate, etc.) who want to do coursework (e.g. project, etc.) in Python.
- Someone who has experience in some other programming language (e.g. C/C++, PHP, Perl, etc.), but has never done programming in ANDROID.
- Someone who is already working as a professional VB.NET developer and wants to switch to ANDROID.
- Someone who did her undergraduate in Economics, has been working in Media sector since graduation, and also working as a professional freelance PHP developer.
- Electrical/Electronic undergraduates in their 3rd semester who want to beef up their software skills prior to graduation.

Prerequisites:

- Successfully complete the entrance test with score of at least 40% (for trainees directly applying to this level).
- Successfully complete the DWIT Training – Programming in Python – Level 1 course (not applicable to trainees directly applying to this level).
- Successfully complete the interview.
- Willing and eager to spend at least 10-20 hours (varying from student-to-student) per week outside of the training class to read/write codes in Python (self-study and practice).

TRAINING METHOD

- The course is spread over 40 hours that consists of approximately 15 hours of lecture and 25 hours of hands-on lab work.
- Lab exercises are mandatory, have a fixed deadline, and are graded. The course puts heavy emphasis on lab exercises because software programming can only be learnt well by explicitly putting into practice the principles that have been taught (i.e. in simpler terms – by doing lots and lots of coding). Late submission (past the deadline) of exercises incur some penalty from total points.
- Instructors may provide relevant lecture/lab notes to students as (and when) necessary in the form of printed handouts and or via emails.
- Instructors may provide supplementary code snippets to students via email or in lab class to support the theory and or lab material that is being taught.
- At the end of the course, students may have to give an exam (which will be optional), that will test their knowledge on the material covered during the course. This exam may be practical and/or theoretical and is mandatory for any student wishing to join a higher level.
- Students are graded on the basis of attendance, lab exercises and exam in the increasing order of importance.

In summary, the only effective way to learn programming is to write lots of code. So, in order to really make this training productive, students are encouraged to spend as much time as necessary to complete the lab exercises on time. As part of the course, students will spend at least 30 hours in the lab but especially if you are new to programming or are coming from a non-computer-science background, it is recommended that you spend at least 10-20 hours per week outside of the class on your own to practice coding in Python.

COURSE DURATION

- 40 hours
- Classes
  - ✓ Morning/Evening

COURSE DURATION

1. WEB APPLICATION BASICS
    - How the web works
    - Overview of django
    - Django philosophies
    - What we are going to build

2. GETTING STARTED
    - Introduction of projects and apps
    - Creating a basic simple web application
    - Introduction to django admin and how to use it
    - How to use applications

3. VIEWS AND TEMPLATES
    - Introduction to views and templates
    - How views and templates work
    - Introduction to bootstrap
    - Bootstrap for basic ui design
    - Introduction to django forms and how to use it

4. MODELS
    - Introduction to models
    - How models work
    - Using databases
    - Making queries
    - Database relationships
    - Sessions

5. RESTFUL API
    - Introduction to rest api
    - Rest api application
    - Creating rest api for application
    - Using rest api in ui

6. EXTRAS
    - CONCLUSION AND DISCUSSIONS ON THE SUBJECT

LABS

Lab assignments will focus on the practice and mastery of contents covered in the lectures; and introduce critical and fundamental problem-solving techniques to the students.

DISCLAIMER

Please note that Deerwalk Training Center reserves the right to change the course syllabus of DTC – Programming in Python – Level 2 course at any time without prior notification.

Programming in Python: Level 3

OBJECTIVES

This course builds on the foundation laid by DTC – Programming in Python – Level 3 to prepare trainees for a career as Python software engineer.

TARGET GROUP

Prerequisites:

- Successfully completed the DWIT Training – Programming in Python – Level 2 or obtained at least 40% score on the entrance exam.
- The latter case applies for new students that are directly attempting this training.
- Successfully complete the interview.
- Willing and eager to spend at least 10-20 hours (varying from student-to-student) per week outside of the training class to read/write codes in Python (self-study and practice).
- Please note that this is a lab intensive course where the students will be expected to work on lab exercises for approximately half the duration of the session.

TRAINING METHOD

- The course is spread over 40 hours that consists of approximately 20 hours of lecture and 20 hours of lab work.
- Lab exercises are mandatory, have a fixed deadline, and are graded. The course puts heavy emphasis on lab exercises because software programming can only be learnt well by explicitly putting into practice the principles that have been taught (i.e. in simpler terms – by doing lots and lots of coding). Late submission (past the deadline) of exercises incur some penalty from total points.
- Instructors may provide relevant lecture/lab notes to students as (and when) necessary in the form of printed handouts and or via emails.
- Instructors may provide supplementary code snippets to students via email or in lab class to support the theory and or lab material that is being taught.
- At the end of the course, students may have to give an exam (which will be optional), that will test their knowledge on the material covered during the course. This exam may be practical and/or theoretical and is mandatory for any student wishing to join a higher level.
- Students are graded on the basis of attendance, lab exercises and exam in the increasing order of importance.

COURSE DURATION

- 40 hours
- Classes
  - ✓ Morning/Evening

COURSE BREAKDOWN

1. INTRODUCTION TO PYTHON FOR DATA SCIENCE
    - Python Basics
    - Different data types

2. PYTHON LISTS
    - Store many different data points under a single name
    - Create, subset and manipulate
    - Lists in all sorts of ways

3. FUNCTIONS AND PACKAGES
    - Importing Python packages
    - Calling functions
    - Numpy
    - Write superfast code with Numerical Python
    - Create different types of visualizations A package to efficiently store
    - Calculations with huge amounts of data

4. MATPLOTLIB
    - Create different types of visualizations
    - Learn how to build Complex and customized plots based on real data.

5. CONTROL FLOW AND PANDAS
    - Write conditional constructs to tweak the execution of scripts
    - The Pandas DataFrame
    - The key data structure for Data Science in Python

6. PYTHON FOR DATA SCIENCE (ADVANCE COURSE)
    - Getting Started with Data Science
    - Data Science: Generating value from Data
    - The Data Science Process
    - Week Assignment

7. BACKGROUND IN PYTHON AND UNIX
    - Key Data Stuctures
    - Week Assignment

8. JUPYTER NOTEBOOKS AND NUMPY
    - Jupyter Notebooks
    - Numpy Advance tutorial practice
    - Satellite Image Application in numpy
    - Week Assignment

9. PANDAS
    - Working with pantaa
    - Week Assignment

10. DATA VISUALIZATION
    - Introduction to Data Visualization
    - Case Studies
    - Matplotlib and other Libraries
    - Week Assignment

11. INTRODUCTION TO MACHINE LEARNING
    - Regression Classification
    - Clustering
    - Analysis
    - Week Assignment

12. WORKING WITH TEXT AND DATABASES
    - Working with Databases
    - Natural Language Processing
    - Working with Text
    - Week Assignment

13. FINAL PROJECT

LABS

Lab assignments will focus on the practice and mastery of contents covered in the lectures; and introduce critical and fundamental problem-solving techniques to the students.

DISCLAIMER

Please note that Deerwalk Training Center reserves the right to change the course syllabus of DTC – Programming in Python – Level 3 course at any time without prior notification.