**Python Training Map**

**Programming World**

Introduction

1. History 2. Programming Paradigm 3. Features 4. Who uses Python today?

Setup Workspace

1. Install python from www.python.org 2. Install Anaconda www.anaconda.com 3. Install Jupyter Notebook

1

Basics

01. Variables 02. Print function 03. Input from user 04. Data Types

a. Numbers b. Strings c. Lists d. Dictionaries e. Tuples f. Sets g. Other Types 05. Operators

a. Arithmetic Operators b. Relational Operators c. Bitwise Operators d. Logical Operators 06. Type conversion

Control Statements

1. If Else a. If

b. Else c. Else If d. If Else Ternary Expression 2. While Loops

a. Nested While Loops b. Break c. Continue d. pass e. Loop else

2

Lists

1. List Basics 2. List Operations 3. List Comprehensions 4. List Methods

Strings

1. String Basics 2. String Literals 3. String Operations 4. String Comprehensions 5. String Methods

For Loops

1. Range Functions 2. Nested For Loops 3. Break 4. Continue 5. Pass 6. Loop else

Functions

1. Definition 2. Call 3. Function Arguments 4. Default Arguments 5. Docstrings 6. Scope 7. Special functions Lambda, Map and Filter 8. Recursion 9. Functional Programming and Reference Functions

3

Dictionaries

1. Dictionaries Basics 2. Operations 3. Comprehensions 4. Dictionaries Methods

Tuples

1. Tuples Basics 2. Tuples Comprehensions 3. Tuple Methods

Sets

1. Sets Basics 2. Sets Operations 3. Union 4. Intersection 5. Difference and Symmetric Difference

Data Structures and Algorithms

1. Analysis of Algorithms 2. Types of analysis 3. Asymptotic Notations 4. Recursion and Backtracking 5. Stack 6. Queue 7. Trees 8. Linked Lists 9. Sorting 10. Searching

4

File Handling

1. File Basics 2. Opening Files 3. Reading Files 4. Writing Files 5. Editing Files 6. Working with different extensions of file 7. With Statements

Exception Handling

1. Common Exceptions 2. Exception Handling

a. Try b. Except c. Try except else d. Finally e. Raising exceptions f. Assertion

Object Oriented Programming

1. Classes 2. Objects 3. Method Calls 4. Inheritance and Its Types 5. Overloading 6. Overriding 7. Data Hiding 8. Operator Overloading

5

Regular Expression

1. Basic RE functions 2. Patterns 3. Meta Characters 4. Character Classes

Modules & Packages

1. Different types of modules 2. Create your own module 3. Building Packages 4. Build your own python module and deploy it on pip

Magic Methods

1. Dunders 2. Operator Methods

CGI Programming

1. Architecture 2. GET 3. POST 4. Cookies 5. Working with files

6

Network Programming

1. Socket 2. Modules 3. Networking methods 4. Client and Server 5. Other Modules

Multithreading

1. Thread 2. New Thread 3. Threading Module 4. Synchronization 5. Priorities

GUI Programming

1. GUI Programming Basics 2. Using Tkinter 3. Building Desktop Applications using Tkinter.

7

Advance Topics

1. Flask 2. SQL 3. HTML5 4. CSS3 5. JavaScript and jQuery 6. Web Scraping 7. Projects

Approach for Academic Requirements

1. All curriculum topics of the respective university will be covered within this training

program. 2. For theory, material will be provided.

Approach for Placement Requirements

1. Competitive Programming

Competitive Programming will start after Loops and Functions, parallelly with further topics.

2. Logical Thinking

Logical Programming will start from day one parallelly.

3. Building Strategy for Technical Rounds 4. Solving more than 500+ problem statements and interview questions. 5. Building mini Projects for understanding of different modules.

8

Further Approach after learning Python Core

1. Web Development 2. Ethical Hacking 3. Machine Learning 4. Deep Learning 5. Computer Vision 6. Natural Language Processing 7. Data Science

Materials & Resources

1. E-books 2. PDFs 3. Booklets 4. Jupyter Files 5. Useful links

Will be provided as we move further with each and every topic.

9