

## COURSE STRUCTURE

| <b>CourseCode</b>             | BCA10060                  |          |                        |         |       |
|-------------------------------|---------------------------|----------|------------------------|---------|-------|
| <b>CourseCategory</b>         | <b>Program Foundation</b> |          |                        |         |       |
| <b>CourseTitle</b>            | <b>Python</b>             |          |                        |         |       |
| <b>Teaching Scheme</b>        | Lectures                  | Tutorial | Laboratory / Practical | Project | Total |
| <b>Weekly load hours</b>      | 2                         | -        | 2                      | -       | 3     |
| <b>Credits</b>                | 2                         | -        | 1                      | -       | 3     |
| <b>Assessment Schema Code</b> | TL4                       |          |                        |         |       |

**Pre-requisites:** Basic understanding of Computer Programming terminologies. A basic understanding of any of the programming languages.

- **Course Objectives:** To understand the concept of Python.

### **Course Outcomes:**

1. On completion of the course, student will be able to—
2. To understand why Python is a useful scripting language for developers.
3. To learn how to design and program Python applications.
4. To learn how to use lists, tuples, and dictionaries in Python programs.
5. To learn how to identify Python object types.
6. To define the structure and components of a Python program.
7. To learn how to write loops and decision statements in Python.

### **Course Contents:**

#### **Unit- I: Introduction to Python [2]**

History

Features

Setting up path

working with Python

Basic Syntax

Variable and Data Types

Operators

#### **Unit- II : Conditional Statements and Loops[6]**

If, If- else, Nested if-else

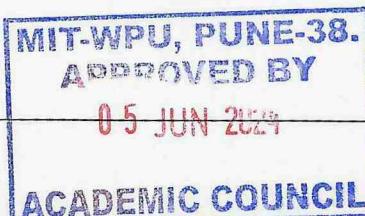
For Loop, While Loop, Nested Loop

Break, Continue, Pass

Using Strings

Accessing Strings

Basic Operations



String slices  
Function and Methods

### Unit III: Lists, Tuple and Dictionaries,Sets[8]

Lists – Introduction

Creating and Accessing elements

Updating and deleting lists,traversing ,reversing a list

Built in list operators ,Function and Methods

Tuples:-Introduction, Accessing tuples, Tuple Assignment, Tuples as return values, Variable-length argument tuples, and Basic tuples operations, Concatenation, Repetition, in Operator, Iteration, Built-in tuple functions, indexing, slicing and matrices. Dictionaries:-Introduction,Creating a Dictionary, Accessing Values in a dictionary, Updating Dictionary, Deleting Elements from Dictionary, Properties of Dictionary keys, Operations in Dictionary, Built-In Dictionary Functions, Built-in Dictionary Methods

Sets- Definition, transaction of set(Adding, Union, intersection), working with sets

### Unit IV: Functions[3]

Defining a function,

calling a function

Types of functions

Function Arguments

Anonymous functions

Global and local variables

### Unit V: Modules and Packages[2]

Importing module

Math module

Random module

Packages Composition

### Overview of OOP[2]

Creating Classes and Objects

Class attributes:-Class and Instance Variables

Class methods,Instance methods,Static Methods

Inheritance in Python

### Unit VI: Python Libraries[7]

#### Numpy

NumPy ndarray - Vectorization Operation - Array

Indexing and Slicing - Transposing Array

and Swapping Axes - Saving and Loading Array - Universal

Functions - Mathematical and

Statistical Functions in Numpy

#### Pandas

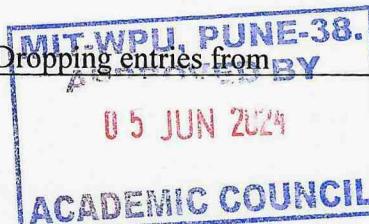
Series and DataFrame data structures in pandas -

Creation of Data Frames – Accessing the

columns in a DataFrame - Accessing the rows in a

DataFrame - Panda's Index Objects -

Reindexing Series and DataFrames - Dropping entries from BY



**Series and Data Frames -**

Indexing, Selection and Filtering in Series and Data Frames

- Arithmetic Operations between

Data Frames and Series - Function Application and

Mapping.

**LearningResources:**

**Reference Books:**

1. Dive into Python by Mark Pilgrim
2. Programming Python by Mark Lutz, O'Reilly Media
3. Python Testing Cookbook by Greg L. Turnquist
4. Head First Programming by Paul Barry and David Griffiths
5. Python Programming: An Introduction to Computer Science by John Zelle

**Supplementary Reading:** Python Programming: An Introduction to Computer Science by John Zelle

**Weblinks:**

<https://docs.python.org/3/tutorial/index.html>

[www.w3schools.com/python/](http://www.w3schools.com/python/)

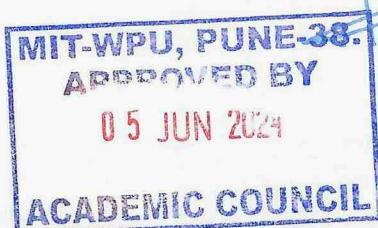
[https://www.tutorialspoint.com/python/index.htm](http://www.tutorialspoint.com/python/index.htm)

**MOOCs:** Online courses for self-learning

Courses by NPTEL and MIT Open Courseware etc

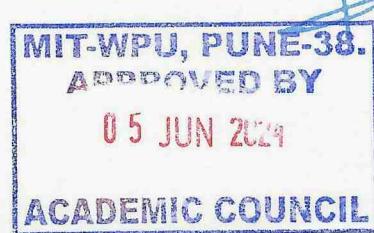
**Pedagogy:**

4. Participative Learning,
5. discussions,
6. algorithm,
7. programming concepts,
8. experiential learning through practical problem solving,
9. assignments.



Dr. B. Dabholkar

|            |                                           |
|------------|-------------------------------------------|
| Sr.<br>No. | Practicals to be conducted on             |
| 1          | <b>Conditional Statements</b>             |
| 2          | <b>Lists, Tuple and Dictionaries,Sets</b> |
| 3          | Functions                                 |
| 4          | Modules and Packages                      |
| 5          | Classes and Objects                       |
| 6          | Python Libraries                          |



EB  
Bellur