ITM SLS
BARODA UNIVERSITY
"Think Big...
Think Beyond"

ITM (SLS) Baroda University
School of Engineering
Department of Computer Science and Engineering

Course Name: Programming in Python-1
Course Type: Core

Teaching Scheme			Credits	Examination Marks				Total Marks
L	T	P	C	Theory Marks		Practical Marks		
				External	Internal	External	Internal	
3	0	4	5	40	60	20	30	150

What is Python?

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991. It is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

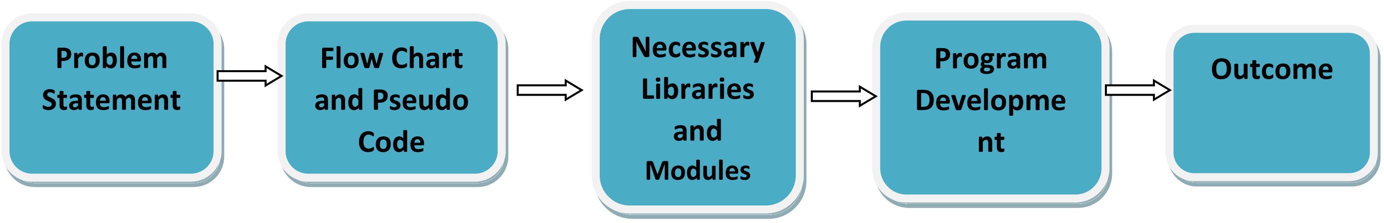
Python was designed for readability, and has some similarities to the English language with influence from mathematics. It has syntax that allows developers to write programs with fewer lines than some other programming languages. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Course overview

In this course, students will learn how to do programming in Python. Students will learn: how Python works and its place in the world of programming languages; to work with and manipulate strings; to perform math operations; to work with Python sequences; to collect user input and output results; flow control processing; to write to, and read from, files; to write functions; to handle exception; and work with dates and times.



Prerequisite

This course does not require any programming background. This course helps the students to learn programming in python.

Learning outcomes:

After completing the course, the student shall be able to :

	Course Outcome	Bloom’s Level
CO1	Understand basics concepts of Python programming	Understanding
CO2	Describe the Numbers, Math functions, Strings, List, Tuples and Dictionaries in Python	Application
CO3	Express different decision making statements and functions	Applying
CO4	Learn implementation essentials of python	Creation
CO5	Implement the specified mini-projects	Creation

Textbook:

Sr.No	Book Name
1	SheetalTaneja, Naveen Kumar, “Python Programming: A Modular Approach”, Pearson (2019)
2	R. NageswaraRao, “Core Python Programming”, dreamtech

Reference Book:

Sr.No	Book Name
1	ReemaThareja, “Python Programming: Using Problem Solving Approach”, Oxford University Press (2017)
2	John V Guttag, “Introduction to Computation and Programming using Python with Application to Understanding Data”, PHI (2016)
3	Martin C Brown, “Python: The Complete Reference”, McGraw-Hill Education (2018)
4	YashavantKanetkar, “Let us Python”, BPB Publication (2019)
5	Stephen Klosterman, “Data Science Projects with Python: A case study approach to successful data science projects using Python, pandas, and scikit”, Packt Publishing (2019) Kindle edition

Required Software:

1. Python Version 3.6 https://www.python.org/downloads/
2. Google Colab: https://colab.research.google.com/notebooks/io.ipynb

Learning Resources:

Learning Related Website: https://www.python-course.eu/

TedEx Videos:

Sr. No	TEDx Video
T1	https://www.youtube.com/watch?v=ENWVRcMGDoU– How algorithms shape our world.- Kevin Slavin Kevin Slavin argues that we're living in a world designed for -- and increasingly controlled by -- algorithms. In this riveting talk from TEDGlobal, he shows how these complex computer programs determine espionage tactics, stock prices, movie scripts, and architecture. Slavin

	also warns that we are writing code we can't understand with implications we can't control.
T2	World Changing: Data Science and AI | Fred Blackburn | TEDxURL The lecture covered the key points like tremendous increase in data, real world examples of machine learning i.eAlexa, Robot Scientist, Healthcare industry, artwork and many more.Video also covered the racing trends of living and working with human intelligence and the learning pattern of human mind and machine

Other Videos:

Sr. No	**About Video**	**Link**	**Topic**
O1	Erik Demaine, Ronald Rivest, and SriniDevadas. 6.006 Introduction to Algorithms. Spring 2008. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu. License: Creative Commons BY-NC-SA	https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-spring-2 008/	Introduction to Algorithm
O2	Lecture by Professor Jerry Cain for Programming Paradigms (CS107) in the Stanford University Computer Science department.	https://www.youtube.com/watch?v=Ps8jOj7diA0&list=PLD28639E2FFC4B 86A	Programming Paradigm
O3	Dr. Anna Bell (MIT 6.0001 Introduction to Computer Science and Programming in Python, Fall 2016)	https://www.youtube.com/watch?v=RvRKT-jXvko&list=PLUl4u3cNGP63WbdFxL8giv4yhgdMGaZNA&index=17	List, Tuple and Dictionary
O4	Introduction to Python by Harvard University (Lecture-06 CS50 2018)	https://www.youtube.com/watch?v=mvlTSMUNQN4&t=1243s	Basics, Data Types, Control Statements

Related MOOCs courses:

Sr.No	MOOC Courses
M1	"The Joy of Computing Using Python" by Prof. SudarshanIyengar, IIT Ropar 12 Weeks on NPTEL.
M2	"Programming for Everybody (Getting Started with Python)",7 week course offered by University of Michigan and Courseera.

Course Outline:

Unit #	Topics	Lab	Assignment	Teaching Hours
1	**Introduction to Programming:** Introduction to Programming Fundamentals, programming environment, principles of programming, what is debugging, text editors and debuggers, introduction to Flow-Chart and Algorithm	P1	--	6
2	**Introduction to Python:** History, Features, Versions, Applications, Setting up path, Installation and Working with Python, Fundamentals of Python, Basic Syntax, Understanding Python variables, rules for naming identifiers and variables in python, operators and expressions, print(), type() and id() functions, taking user input using input() and raw_input() functions. **Data Types:** Integer, Float, complex numbers, Concept of Mutable and Immutable, String Manipulation.	P1	A1	6
3	**Data Types:** Lists, Tuples, Sets, Dictionaries, working with data types and their in-built functions, Logical Constructs, Boolean expressions, Looping, Python for loop, Python range(), Python Nested Loop Structures, Iteration, If-else, while loops, Break-Continue, Pass.	P2,P3, P4,P5, P6	A1,A2	8
4	**Functions and Recursion:** Defining a function, calling a function, Types of functions, Function Arguments, Anonymous functions, Passing Collections to a Function, Keyword and Optional Parameters, Local and global variables, Function as Function arguments, Defining recursion and its application, programming through recursion	P7,P8, P9,P10	A1,A3	8
5	**Modules:** Importing Module, The Math Library, Random Numbers, Sys Module, OS, Date and Time module with their in-built functions	P11	A3	8

6	**File Handling:** File creation, open() and close() methods, read() and write() methods, file modes, file encoding, file object attributes, renaming and deleting files, Knowing Whether a File Exists or Not, Working with Binary Files, Appending Text to a File, Reading Text Files, File Exceptions, The with Statement, Python directory, directory methods and functions.	P12	--	8
7	**Introduction to Object Oriented Python:** Class, Object, Defining variables and functions within a Class	P13	--	2
	TOTAL			**46**

Lab Experiments:

Sr. No.	**Program Statement**
P1	1. Write a Python program to calculate the addition of two numbers without using third variable. 2. Demonstrate the use of id(), type() and size() function in python. 3. Write a Python program to perform basic operations in python.
P2	1. Write a Python program to find greatest among three numbers entered by user. 2. Write Python program to check whether the entered number by user is even or odd. 3. Write a Python program to check whether the entered year is leap year or not. 4. Write a Python program to find all prime numbers within a given range.
P3	Write a Python program to take marks of 5 subjects from user and print obtained grade. (>=90% - A+, 70%-90% - A, 60%-70% - B+, 50%-60% - B, 35%-50% - C, <35% - F)
P4	Write a Python program to implement a simple calculator.
P5	1.Write a Python program for various functions of string in python. 2. To add 'ing' at the end of a given string (length should be at least 3). If the given string already ends with 'ing' then add 'ly' instead. If the string length of the given string

	5. is less than 3, leave it unchanged. Sample String : 'abc' Expected Result : 'abcing' Sample String : 'string' Expected Result : 'stringly'
P6	1. Write a Python program for list and its various in-built methods. 2. Write a Python program to calculate the sum of numbers stored in a list. 3. Write a Python program for tuple and its various in-built methods. 4. Write a Python program for dictionary and its various in-built methods. 5. Write a Python program for set and its various in-built methods.
P7	Write a function that takes a list of numbers as input from user and produces the corresponding cumulative list where each element at index i is the sum of elements at index j <= i. For example, Input List = [3, 5, 2, 7, 9, 4]; the Output List = [3, 8, 10, 17, 26, 30].
P8	Write a Python program that takes a sentence as input from the user and computes the frequency of each letter. Use a variable of dictionary type to maintain the count.
P9	Write a function that takes a number as an input argument and returns the corresponding text in words, for example, if input is 368, the function should return ‘Three’, ‘Six’, ‘Eight’. Use a dictionary for mapping digits to their string representation.
P10	1. Write a Python program to find factorial of a given number using recursion 2. Write a Python program to print ‘n terms of Fibonacci series using recursion.
P11	1. Write a Python program to perform various functions of math module. 2. Write a Python program to perform various functions of random module. 3. Write a Python program to perform various system related functions using sys and os module. 4. Write a Python program to retrieve the date and time related information of a system using python.
P12	1. Write a Python program that reads a text file and changes the file by capitalizing each character of file.

	2. Write a Python program to append data to an existing file 'python.py'. Read data to be appended from the user. Then display the contents of entire file. 3. Read a text file in Python and print no. of lines and no. of unique words.
P13	Write a Python Program to create a class “Rectangle” with variables length and width. Create three different objects for a class and write a function that displays the area of a rectangle.

Assignments:

Sr. No.	**Assignment Name**
A1	**Guess the Number** The Goal: This assignment also uses the random module in Python. The program will first randomly generate a number unknown to the user. The user needs to guess what that number is. (In other words, the user needs to be able to input information.) If the user’s guess is wrong, the program should return some sort of indication as to how wrong (e.g. The number is too high or too low). If the user guesses correctly, a positive indication should appear. You’ll need functions to check if the user input is an actual number, to see the difference between the inputted number and the randomly generated numbers, and to then compare the numbers. Concepts to keep in mind: Random function Variables Integers Input/Output Print While loops If/Else statement

A2	Perform the following task using dictionary for the campus drive of a company which required 75 minimum CGPI criteria for appearing in the interview: a) Create the dictionary for department of CSE, Automobile, Mechanical, Civil and Electrical with name as key and their current CGPI with values. b) Combine all the department detail into one common dictionary and named it as ITM. c) Display the total number of students of university who are eligible for the drive. d) Display the total number of students of university who are not eligible for the drive.
A3	**Problem: An advanced Math Learning Tool** The program will generate just one question for each run that generate five questions and after a student answer all five, report the number of correct answers. The program should also display the time spent on test and list all the questions. **Sample Input and Output:** What is 9-2? 7 You are correct What is 3-0? 3 You are correct What is 3-2? 1 You are correct What is 7-4? 4 You answer is wrong 7-4 should be 3 What is 7-5? 3 You answer is wrong 7-5 should be 2 Correct count is: 03

	Test time is 1021 seconds 9-2=7 Correct 3-0=3 Correct 3-2=1 Correct 7-4=4 Wrong 7-5=4 Wrong