LOK JAGRUTI UNIVERSITY (LJU)

INSTITUTE OF ENGINEERING & TECHNOLOGY

Department of Civil Engineering (709)

Bachelor of Engineering (B.E.) – Semester – I

Course Code:	**017092191**
Course Name:	**Python Programming**
Category of Course:	Engineering Science Course (ESC)
Prerequisite Course:	Mathematics - I (017091191)

Teaching Scheme				
Lecture (L)	**Tutorial (T)**	**Practical (P)**	**Credit**	**Total Hours**
2	**0**	**6**	**5**	**20**

Syllabus

Unit No.	Topic	Prerequisite Topic	Successive Topic	Teaching Hours
01	**Introduction to Python and Jupyter Notebooks**			**1 (5%)**
	1.1 Overview of languages : introduction and comparison of procedure, object oriented and machine level language, introduction, uses and features of python, difference of compiler and interpreter, use function of print	---	---	
	1.2 Jupyter Notebooks: creating, opening, saving and downloading notebooks	---	---	
	1.3 using interactive shell, editing, saving and running a script, basics of python IDLE, IDEs, IDE	---	---	
02	**Basic Elements of Python**			**2 (10%)**
	2.1 Basics and variables of data types for text, numeric and boolean (text, numeric, sequence, mapping, set, Boolean, range and len function), data type conversion and single line and multiline comments	---	---	
	2.2 Arithmetic, relational, logical, ternary, bitwise, assignment, identity and membership operators and their expressions, operator precedence	---	---	
	2.3 Reading input from users for text and numeric	---	---	
03	**Decision Making Statements**			**2 (10%)**
	3.1 Control Statements: Simple if, if-else, if-elif-else(ladder if), Nested if	---	---	
04	**Looping Constructs**			**3 (15%)**
	4.1 Loops: for loop, while loop, nesting of loops	---	---	
	4.2 Break, continue, pass statement	---	---	
05	**Functions and Scoping**			**2 (10%)**
	5.1 Declaring, defining and invoking different categories of user define functions.	---	---	
	5.2 One liner and multi liner function specification	---	---	
	5.3 Function arguments: keyword, default, positional and arbitrary or variable-length	---	---	
	5.4 Local v/s Global variables, modules	---	---	
06	**Immutable Data Structure**			**2 (10%)**
	6.1 Immutable data structures (operations and functions): strings, tuples, numbers	---	---	
	6.2 Strings -immutability, declaring, creating, accessing character of string by index and slice operator slicing, mathematical operators, comparison, joining and formatting, removing spaces and changing cases of string. function of len, find, count, replace, partition and split of string. check type of character presence in string using istitle(), isalnum(), islower(), isupper(), isnumeric(), isalpha(), isdigit(), isidentifier(), ispritable(), isspace()	---	---	
	6.3 Tuples - immutability, creating, accessing elements by index and slice operator slicing and mathematical operator, function of len, count, index, sorted, min, max, tuple packing and unpacking	---	---	
07	**Mutable Data Structure**			**3 (15%)**
	7.1 List- mutability, creating, accessing elements by index and slice operator slicing, operators - mathematical, comparison, membership, functions - len, count, index, append, insert, extend, remove, pop, clear, sort, reverse, split, aliasing and cloning of list – slice and copy, nested list, nested list as matrix, list comprehensions	---	---	
	7.2 Set - mutability, creating and accessing element of set, mathematical operations – union, intersection, difference, symmetric difference, membership operators functions – len, add, update, copy, pop, remove, discard, clear, set comprehensions	---	---	
	7.3 Dictionaries- mutability, creating, accessing, updating, deleting elements of dictionary functions – dict, len, clear, copy, update, get, popitem, keys, values, items, setdefault, dictionary comprehensions, loop and nested dictionaries	---	---	
	7.4 Lambda function, lambda function with map, reduce and filter	---	---	

08	**Working with Files**			**2 (10%)**
	8.1 Types of files	---	---	
	8.2 Create, open, with open, read (read(), readlines(), readline()) , write((write(), writelines()), append (tell(), seek(offset)), close, rename or delete text files. various properties of file object.	---	---	
	8.3 OS module- getting and changing current working directory, creating, removing and renaming directories, list files and sub directories	---	---	
	8.4 System modules – sys.argv, sys.exit, sys.maxsize, sys.path, sys.version	---	---	
09	**Mathematical Functions in Python**			**1 (5%)**
	9.1 Import math module	---	---	
	9.2 ceil, comb, floor, exp, fabs, factorial, log, pow, fmod, frexp	---	---	
	9.3 Trigonometric, logarithmic functions, maths constants	Trigonometric Function (017091191-Unit-2)	---	
10	**The Matplotlib Library**			**2 (10%)**
	10.1 Installation and import of matplotlib and numpy	---	---	
	10.2 Function for graph: create label, title, legend, set font properties, grid, plot, show, subplot, color, colormap	---	---	
	10.3 Line graph: line style, marker, marker size, format string fmt, color reference, multiple line, two lines on same graph, twinx and twiny function	---	---	
	10.4 Scatter graph: marker style, color and size, alpha	---	---	
	10.5 Bar graph: horizontal and vertical bar, width, height, color	---	---	
	10.6 Histogram and Box Plot	---	---	
	10.7 Pie chart: lables, array, color, startangle, explode, shadow	---	---	
	10.8 plotting maths functions			

Sr No.	Practical Title	Link to Theory Syllabus
1	Write a Python program to add 2 Numbers with user input.	Unit-2
2	Write a Python program to find the area of Circle.	Unit-2
3	Write a Python program to find the area of Triangle.	Unit-2
4	Write a Python program to calculate the area of a trapezoid.	Unit-2
5	Write a Python program to calculate surface volume and area of a cylinder.	Unit-2
6	Write a Python program to convert Fahrenheit to Celsius.	Unit-2
7	Write a Python program to convert hours into minutes and seconds.	Unit-2
8	Write a Python program to calculate the square root of a positive number.	Unit-2
9	Write a Python program to calculate the square root of a complex number.	Unit-2
10	Write a Python program to find roots of quadratic equation.	Unit-2
11	Write a Python program to convert degree to radian.	Unit-2
12	Write a program to check if the input number is odd or even. (Simple if).	Unit-3
13	Write a program to find the maximum number among the three input numbers.	Unit-3
14	Write a program to check if year is a leap year or not (Nested If).	Unit-3
15	Write a program to find sum of first N natural number given by user.	Unit-4
16	Write a python program to read three numbers (a,b,c) and check how many numbers between ‘a’ and ‘b’ are divisible by ‘c’.	Unit-4
17	Write a Python program that prints all the numbers from 0 to 6 except 3 and 6.	Unit-4
18	Write a program to find the factorial of a number provided by the user	Unit-4
19	Write a python program to display the Fibonacci sequence up to n-th term.	Unit-4
20	Write a program to take 10 values from keyboard using loop and print their average on the screen	Unit-4
21	Write a program to reverse a number.	Unit-4
22	Write a program to check whether a number is Armstrong number or not.	Unit-4
23	Write a program to check if a number is prime or not.	Unit-4
24	Write a program to print prime number between given interval from user.	Unit-4
25	Write a Python Program to read a number n and print an identity matrix of the desired size.	Unit-4

26	Draw a pattern:
	* * * *
	  * * *
	  * *
	  *
		Unit-4

27	Draw a pattern:
	  *
	  * *
	  * * *
	* * * *
		Unit-4

28	Draw a pattern: * * * * * * * * * *	Unit-4
29	Draw a pattern: * * * * * * * * * *	Unit-4
30	Draw a pattern: 1 2 3 4 5 1 2 3 4 1 2 3 1 2 1	Unit-4
31	Write a Python function to find the Max of TWO numbers.	Unit-5
32	Write a Python function to sum all the numbers in a list.	Unit-5
33	Write a Python function to calculate the factorial of a number.	Unit-5
34	Write a Python function to check whether a number is in a given range.	Unit-5
35	Write a Python program to read an entire text file.	Unit-6
36	Write a python program to write a list to a file	Unit-6
37	Write a Python program to count the number of lines in a text file.	Unit-6
38	Write a Python program to check if a string is palindrome or not	Unit-7
39	Write a Python program to Find length of a string in python.	Unit-7
40	Write a Python program to split and join a string	Unit-7
41	Write a Python function that accepts a string and calculate the number of uppercase letters and lowercase letters.	Unit-7
42	Write a Python program to Convert Snake case to Pascal case.	Unit-7
43	Write a Python program to demonstrate the addition of elements in a Tuple.	Unit-7

44	Write a Python program to demonstrate the negative index in a Tuple	Unit-7
45	Write a Python program to demonstrate the slicing of a Tuple	Unit-7
46	Write a Python program to print the even numbers from a given list.	Unit-8
47	Write a Python Program to print the largest even and largest odd number in a list.	Unit-8
48	Write a Python program to swap first and last element of the list.	Unit-8
49	Write a Python program to find the sum of all the elements in the list.	Unit-8
50	Write a Python program of Reversing a List.	Unit-8
51	Write a Python program to Merging two Dictionaries	Unit-8
52	Write a Python program for Words Frequency in String Shorthand's.	Unit-8
53	Write a Python program to calculate the sum of the positive and negative numbers of a given list of numbers using lambda function.	Unit-8
54	Write a Python program to rearrange positive and negative numbers in a given array using Lambda.	Unit-8
55	Write a Python program to count the even, odd numbers in a given array of integers using Lambda.	Unit-8
56	Write a Python program to add two given lists using map and lambda.	Unit-8
57	Write a Python program to find numbers divisible by nineteen or thirteen from a list of numbers using Lambda.	Unit-8

Major Components/ Equipment	
Sr. No.	**Component/Equipment**
1	Computer
2	Python Compiler - Pycharm

Proposed Theory + Practical Evaluation Scheme by Academicians
(% Weightage Category Wise and it's Marks Distribution)

L:	**2**	**T:**	**0**	**P:**	**6**

Note: In Theory Group, Total 4 Test (T1+T2+T3+T4) will be conducted for each subject.
Each Test will be of 25 Marks.
Each Test Syllabus Weightage: Range should be 20% - 30%

Group (Theory or Practical)	Group (Theory or Practical) Credit	Total Subject Credit	Category	% Weightage	Marks Weightage
Theory	**2**	**5**	MCQ	24%	60
Theory			Theory Descriptive (Mainly Programming)	16%	40
Theory			Formulas and Derivation	0%	0
Theory			Numerical	0%	0
Expected Theory %	**40%**		**Calculated Theory %**	**40%**	**100**
Practical	**3**		Individual Project	24%	40
Practical			Group Project	24%	40
Practical			Internal Practical Evaluation (IPE)	12%	20
Practical			Viva	0%	0
Practical			Seminar	0%	0
Expected Practical %	**60%**		**Calculated Practical %**	**60%**	**100**
Overall %	**100%**			**100%**	**200**

Course Outcome	
	Upon completion of the course students will be able to
CO1	Understand the basics of python programming.
CO2	Apply the fundamental python concepts such as data types, identifiers, keywords, constants, variable, comment, basic input output, operators, and its precedence.
CO3	Analyze the indentation syntax, branching and looping techniques, and various data structures such as strings, arrays, lists, tuples, dictionaries and sets.
CO4	Apply mathematical functions in python and generate different types of the plots using library.
Suggested Reference Books	
1	Python: The Complete Reference, Martin C. Brown, McGraw Hill Education
2	Introduction to Computation and Programming Using Python, John V Guttag, Prentice Hal
3	Data Structures and Algorithms in Python, Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, Wiley
4	Fundamentals of Python – First Programs, Kenneth A. Lambert, CENGAGE Publication
5	Professional Python, Luke Sneeringer, Wrox

List of Open Source Software/Learning website	
1	www.python.org
2	www.w3schools.com
3	www.geeksforgeeks.org
4	www.learnpython.org

Practical Project/Hands on Project		
Sr. No.	**Project List**	**Linked with Unit**
1	Hotel management system : Listed below are some of the important functions **dashboard()** – This function displays the menu or welcome screen to perform different hotel booking activities mentioned below. **new_acc()** – This function creates a new customer account. It asks for some personal and banking details of the customer such as name, date of birth, citizenship number, address and phone number. **room_type()** – This function allows the user to select the categories of the room ie normal or executive with the option of Ac room or non ac room. **check_availability()** – This functionality allows the user to check the number of room vacant prior booking. **book_room()** – This function allows the user to book the selected room. **search_facilities()** – With this function, if the user selects the executive room than user can search for the extra facilities provided like games, swimming, food service in rooms while booking. **payment()** – This function allows making payment of booked room based on number of days the room is occupied via online method option or at the checkout time.	Unit 2,3,4,6
2	It is required to maintain and process the status of total 9 resources. The status value is to be stored in an integer array of dimension 3x3. The valid status of a resource can be one of the 3 followings: **free**: indicated by integer value 0 **occupied**: indicated by integer value 1 **inaccessible**: indicated by integer value 2 Declare a class called ResourcesStatus, having data member called statusRef, referring to a two dimensional array (3x3) of integers to be used to refer to the above mentioned status values. Define a member method called processStausCount that counts and displays total number of free resources, total number of occupied resources and total number of inaccessible resources. The exception to be raised and handled if total number of occupied resources exceeds total number of free resources. The handler marks status of all inaccessible resources as free. Accept initial status values from user and initialize the array. Raise and handle user defined exception if invalid status value given.	Unit 2,3,4,6,8
3	Create an application that performs the following task associated with the files : 1) Eliminating repeated lines from the files. 2) Reverse the content of file and store in another file. 3) Remove the lines starting from any prefix. 4) Obtain the line number where the particular word is present. 5) Obtaining number of words, characters, white spaces and lines present in that particular file.	Unit 2,3,4,5,6
4	Implement calculator functionality.	All Units
5	Write a program to implement Quadratic equation.	All Units