

## COURSE CONTENT

PYTHON PROGRAMMING LAB								
II Semester: CE / ME								
III Semester: CSD / CSE / CSM								
Course Code	Category	Hours / Week			Credits	Maximum Marks		
25X0575	Core	L	T	P	C	CIA	SEE	Total
		0	0	2	1	40	60	100
Contact Classes: Nil	Tutorial Classes: Nil	Practical Classes: 30			Total Classes: 30			
Prerequisites: Programming for Problem Solving Lab								

### Course Overview:

This Python Programming Laboratory introduces students to fundamental and advanced Python concepts through hands-on experiments. The course covers core programming constructs, data structures, functions, file handling, modules, exception handling, object-oriented programming, and GUI development. It also provides exposure to scientific libraries, digital logic simulation, and real-world problem-solving using Python.

### Course Objectives:

1. To install and run the Python interpreter
2. To learn control structures.
3. To Understand Lists, Dictionaries in python
4. To Handle Strings and Files in Python

### Course Outcomes: After Completion of the Course, Students should be able to

1. Apply Python language fundamentals, interpreter features, control structures, functions, recursion, and modules for solving computational problems.
2. Develop Python programs using lists, tuples, arrays, dictionaries, and strings for data manipulation and validation tasks.
3. Implement file handling, exception handling, and text processing techniques for data analysis and information retrieval.
4. Design object-oriented Python programs using classes, attributes, and methods including graphical object representation and GUI components.
5. Utilize Python libraries and packages such as NumPy, SciPy, Plotting tools, and logic gate simulations for scientific computing and visualization.

### List of Experiments

1.
  - I. Use a web browser to go to the Python website <http://python.org>. This page contains information about Python and links to Python-related pages, and it gives you the ability to search the Python documentation.
  - II. Start the Python interpreter and type `help()` to start the online help utility.
2. Start a Python interpreter and use it as a Calculator.
3. Write a program to calculate compound interest when principal, rate and number of periods are given.
4. Read the name, address, email and phone number of a person through the keyboard and print the details.
5. Print the below triangle using for

4 4  
3 3 3  
2 2 2 2  
1 1 1 1 1

6. Write a program to check whether the given input is digit or lowercase character or uppercase character or a special character (use 'if-else-if' ladder)
7. Python program to print all prime numbers in a given interval (use break)
8. Write a program to convert a list and tuple into arrays.
9. Write a program to find common values between two arrays.
10. Write a function called palindrome that takes a string argument and returns True if it is a palindrome and False otherwise. Remember that you can use the built-in function len to check the length of a string.
11. Write a function called is\_sorted that takes a list as a parameter and returns True if the list is sorted in ascending order and False otherwise.
12. Write a function called has\_duplicates that takes a list and returns True if there is any element that appears more than once. It should not modify the original list.
13. Write a function called remove\_duplicates that takes a list and returns a new list with only the unique elements from the original. Hint: they don't have to be in the same order.
14. The wordlist I provided, words.txt, doesn't contain single letter words. So you might want to add "I", "a", and the empty string.
15. Write a python code to read dictionary values from the user. Construct a function to invert its content. i.e., keys should be values and values should be keys.
16. Add a comma between the characters. If the given word is 'Apple', it should become 'A,p,p,l,e'
17. Remove the given word in all the places in a string?
18. Write a function that takes a sentence as an input parameter and replaces the first letter of every word with the corresponding upper case letter and the rest of the letters in the word by corresponding letters in lower case without using a built-in function?
19. Writes a recursive function that generates all binary strings of n-bit length
20. Write a python program that defines a matrix and prints
21. Write a python program to perform multiplication of two square matrices
22. How do you make a module? Give an example of construction of a module using different geometrical shapes and operations on them as its functions.
23. Use the structure of exception handling all general-purpose exceptions.
24. Write a function called draw\_rectangle that takes a Canvas and a Rectangle as arguments and draws a representation of the Rectangle on the Canvas.
25. Add an attribute named color to your Rectangle objects and modify draw\_rectangle so that it uses the color attribute as the fill color.
26. Write a function called draw\_point that takes a Canvas and a Point as arguments and draws a representation of the Point on the Canvas.
27. Define a new class called Circle with appropriate attributes and instantiate a few Circle objects. Write a function called draw\_circle that draws circles on the canvas.
28. Write a python code to read a phone number and email-id from the user and validate it for correctness.
29. Write a Python code to merge two given file contents into a third file.
30. Write a Python code to open a given file and construct a function to check for given words present in it and display on found.



occurrences

32. Write a function that reads a file file1 and displays the number of words, number of vowels, blank spaces, lower case letters and uppercase letters.
33. Import numpy, Plotpy and Scipy and explore their functionalities.
34. Install NumPy package with pip and explore it.
35. Write a program to implement Digital Logic Gates – AND, OR, NOT, EX-OR
36. Write a GUI program to create a window wizard having two text labels, two text fields and two buttons as Submit and Reset.

### TEXT BOOKS:

1. Supercharged Python: Take your code to the next level, Overland
2. Learning Python, Mark Lutz, O'reilly

### REFERENCE BOOKS:

1. Python Programming: A Modern Approach, Vamsi Kurama, Pearson
2. Python Programming A Modular Approach with Graphics, Database, Mobile, and Web Applications, Sheetal Taneja, Naveen Kumar, Pearson
3. Introduction to Python Programming, Gowrishakar S, Veena A, CRC Press
4. Programming with Python, A User's Book, Michael Dawson, Cengage Learning, India Edition
5. Python for Data Science, Dr. Mohd Abdul Hameed, Wiley publications
6. Core Python Programming, Dr. R. Nageswara Rao, Dreamtech press
7. Introduction to Python, Gowrishankar S, Veena A., CRC Press

### ELECTRONIC RESOURCES:

1. <https://python-iitk.vlabs.ac.in/exp/built-in-functions/simulation.html>
2. <https://python-iitk.vlabs.ac.in/exp/loops/simulation.html>
3. <https://python-iitk.vlabs.ac.in/exp/built-in-modules/simulation.html>
4. <https://python-iitk.vlabs.ac.in/exp/file-operators/simulation.html>
5. <https://python-iitk.vlabs.ac.in/exp/strings/simulation.html>

### MATERIALS ONLINE:

1. Lab Manual

Open-ended experiments