PYTHON PRACTICAL INDEX

SR. NO		PROGRAMME NAME	DATE	SIGN
1.		**Python installation and configuration with windows and Linux**		
2.		**Programs for understanding the data types, control flow statements, blocks and loops**		
	2.1	Write a python Programs to understanding the data types of pythons.		
	2.2	Write a python program accept number from user to check user entered number is even or odd using if-else statement		
	2.3	Write a python program to find a greater number between two numbers using Nested if-else statement		
	2.4	Write a program to accept student roll no, name and five subject mars. print student result like roll no, name, five subject marks, subject total, percentage and grade using if else elif statement.		
	2.5	Write a python program to check whether given number is Armstrong or not using while loop. Accept number from user		
	2.6	Write a python program to check whether given number is palindrome or not using while loop. Accept number from user		
	2.7	Write a program to print prime numbers in between a range.		
	2.7	Print the following pattern 5 4 3 2 1 4 3 2 1 3 2 1 2 1 1		
3.		**Programs for understanding functions, use of built in functions, user defined functions**		
	3.1	Write a program to implement any five built-in functions.		
	3.2	Write a function that takes single character and print “Given character is Vowel”. If it is vowel, “Given character is not vowel” otherwise		
	3.3	Write a python program function that interchange the first and last characters of a given string. (accept input from user)		
	3.4	Write a python program to calculate factorial of a given user accepted number using recursive function.		
	3.5	Write a python program to generate the Fibonacci series using recursive function.		
	3.6	Write Python code to illustrate anonymous function lambda in Python.		
	3.7	Python code to illustrate Decorators with parameters in Python.		

	3.8	Python code to illustrate Generator in Python.		
4.	**Programs to use existing modules, packages and creating modules, packages**			
	4.1	Write a program to return a random item from a list, tuple and string using random module.		
	4.2	Write a python program to display current date and time.		
	4.3	Create a User define module **arithmaticsOperation**. The module having functions like sum, average, power, division, multiplication and return a result. Import this module and execute module functions.		
	4.4	Create a package **MyPackageDemo**. Define two modules like **message.py** and **calculation.py** within a package. Import given package and show desirable output.		
5.	**Programs for implementations of all object-oriented concepts like class, method, inheritance, polymorphism etc. (Real life examples must be covered for the implementation of object-oriented concepts)**			
	5.1	Create a class **employee** with data members: name, department and salary. Create suitable methods for reading and printing employee information.		
	5.2	Write a python program to illustrate a non-parameterized / default constructor and parameterized constructor.		
	5.3	Create **Shape** parent class and **Rect** is the child class. In parent class, we have one function to **display()** which displays the **length** and **breadth** of the rectangle. And in the child class, we have defined the function **area()** to find the area of the rectangle and print the output.		
	5.4	Create one base class named **Shape** and two child classes namely **Rect** and **Tri**. Both classes inherit the **display()** function from the Shape class to print the length and breadth. In **Rect** class, we find the area of the rectangle while in **Tri** class we find the area of the triangle using two different functions **R_area()** and **T_area().**		
	5.5	Write a python program to demonstrate a super().		
	5.6	Write a python program to overload + Operator.		
	5.7	Write a python program to demonstrate a function overloading.		
	5.8	Write a python program to demonstrate a implementation static method.		
	5.9	Write a python class and delete object of that class using del keywords.		
6.	**Programs for parsing of data, validations like Password, email, URL, etc.**			

	6.1	Write a python program for password validation using Regular Expression, consider following condition: • Minimum length of password: 8 characters. • At least one alphabet must between [a-z]. • At least one alphabet should be of Upper Case [A-Z]. At least 1 number or digit between [0-9]. • At least 1 character from [$, @, #].		
	6.2	Python program to check if the string is a valid E-mail address or not using regex.		
	6.3	Python program to check if an URL is a valid or not using Regular Expression.		
7.	**Programs for Pattern finding should be covered.**			
	7.1	Write a program to accept input from the user contains only letters, spaces. Any other character is not allowed, using compile function.		
	7.2	Write a program to accept any string as input from the user to match particular word in giver string using match function.		
	7.3	Write a program to accept any string as input from the user to Search particular word in giver string using search function.		
	7.4	Write a program to demonstrate findall(), split() and sub() function		
8.	**Programs covering all the aspects of Exception handling, user defined exception, Multithreading should be covered.**			
	8.1	Program to check for ZeroDivisionError Exception.		
	8.2	Program to check for ValueError Exception.		
	8.3	Write a python programme to raise a user defined exception if age is less than 18		
	8.4	Write a python program for synchronizing the thread.		
	8.5	Write multithread program where one thread prints square numbers and another thread prints cube of numbers		
9.	**Programs demonstrating the IO operations like reading from file, writing into file from different file types like data file, binary file, etc.**			
	9.1	Write a python program to get the file size of a plain text file		
	9.2	Write a python program to read config file and display the contents of entire file		
	9.3	Write a python program to read first.txt file and display the contents of entire file		
	9.4	Write a python program to read binary file and display the contents of entire file		
	9.5	Programme to open a file in read mode and print number of occurrences of character 'a'		

10.	**Programs to perform searching, adding, updating the content from the file.**			
	10.1	Program to create a simple file and write some content in it.		
	10.2	Write a program to read contents of “first.txt” file and write same content in “second.txt” file.		
	10.3	Write a python program to append data to an existing file ‘python.txt. Read data to be appended from the use, then display the contents of entire file.		
	10.4	Write a python program to read a text file and print number of lines, words and characters.		
	10.5	Write a python program that reads a text file and changes the file by capitalizing each character of file.		
11.	**Program for performing CRUD operation with MongoDB and Python**			
12.	**Basic programs with NumPy as Array, Searching and Sorting, date & time and String handling**			
	12.1	Write a NumPy program to convert a list of numeric values into a one-dimensional NumPy array.		
	12.2	Write a NumPy program to create a 3x3 matrix with values ranging from 2 to 10.		
	12.3	Write a NumPy program to create a structured array from given student name, height, class and their data types. Now sort the array on height.		
	12.4	Write a NumPy program, Find the indexes from given array where the values are odd.		
	12.5	Write a NumPy program to get the dates of yesterday, today and tomorrow.		
	12.6	Write a NumPy program to concatenate element-wise two arrays of string.		
	12.7	Write a NumPy program to capitalize the first letter, lowercase, uppercase, swapcase, title-case of all the elements of a given array.		
13.	**Programs for series and data frames should be covered.**			
	13.1	Write a Pandas program to add, subtract, multiple and divide two Pandas Series.		
	13.2	Write a Pandas program to compare the elements of the two Pandas Series.		
	13.3	Write a Pandas program to create a dataframe from a dictionary and display it.		
	13.4	Write a Pandas program to create and display a DataFrame from a specified dictionary data which has the index labels.		
14.	**Programs to demonstrate data pre-processing and data handling with data frame**			
	14.1	Write a Pandas program to add summation to a row of the given excel file.		

	14.2	Write a Pandas program to print a concise summary of the dataset.		
	14.3	Write a Pandas program to detect missing values of a given DataFrame. Display True or False.		
	14.4	Write a Pandas program to Delete columns from DataFrame using Pandas.drop().		
15.	**Program for data visualization should be covered.**			
	15.1	Write a Pandas program to create a line plot of the historical stock prices of Alphabet Inc. between two specific dates.		
	15.2	Write a Pandas program to create a histograms plot of opening, closing, high, low stock prices of Alphabet Inc. between two specific dates.		
	15.3	Write a Pandas program to create a stacked histograms plot of opening, closing, high, low stock prices of Alphabet Inc. between two specific dates.		
	15.4	Write a Pandas program to create a bar plot of the trading volume of Alphabet Inc. stock between two specific dates.		