

ASSIGNMENT-1

1.	Write a Python Program to print “Hello World” using Script Mode and Interactive Mode.
2.	Write a Python Program to input Name, Roll No and Branch from a user and display it.
3.	Write a Python program that takes a user's name as input and prints a greeting message using string formatting. For example, if the user enters "John," the program should output "Hello, John!"
4.	Write a Python program that calculates the sum of a series of integers entered by the user. The user should be able to input numbers until they enter a specific sentinel value (e.g., 0), and then display the sum.
5.	Write a Python program that converts an integer to a float and vice versa. Display the data type before and after conversion.
6.	Write a Python program that demonstrates the use of logical operators (and, or, not) with boolean values to simulate real-world scenarios.
7.	Write a Python program to declare and initialize variables to store the following information: your name, age, and favorite programming language.
8.	Write a Python program that takes two numbers as input and performs addition, subtraction, multiplication, and division operations. Display the results with appropriate messages.
9.	Write a Python program that converts a temperature in Celsius to Fahrenheit. Prompt the user to enter a temperature in Celsius and then display the result. Ensure proper formatting of the output.
10.	Write a Python program that swaps the values of two variables with and without using a temporary variable.

ASSIGNMENT-2

1.	Write a Python program that prompts the user to enter the coefficients of a quadratic equation (a, b, and c) and solves the equation, displaying the roots (real or complex) as output.
2.	Write a Python program that asks the user for their age and uses conditional statements (if, elif, else) to determine if they are a child, teenager, adult, or senior citizen.
3.	Write a Python program that checks if a given year is a leap year or not. Use conditional statements to determine leap year criteria.
4.	Write a Python program that create a menu-driven program that allows the user to perform various operations (e.g., add, subtract, multiply, divide). Use conditional statements to navigate between options.
5.	Write a Python program that generate and display the multiplication table of a given number using a loop. Allow the user to specify the number.
6.	Develop a Python program that calculates the factorial of a non-negative integer entered by the user using a while loop.
7.	Write a Python program to generate and display the first N terms of the Fibonacci sequence using a for loop. Allow the user to specify the value of N.
8.	Write a Python program that checks if a given integer is prime or not.
9.	<p>Write a python program to check the validity of a password given by the user. The password should satisfy the following criteria:</p> <ul style="list-style-type: none">a) Contain at least 1 letter between a and zb) Contain at least 1 number between 0 and 9c) Contain at least 1 letter between A and Zd) Contain at least 1 character from \$, #, @e) Minimum length of password: 6f) Maximum length of password: 12 <p>Based on above criteria checks password strength (weak, moderate, strong).</p>
10.	<p>Given an input file which contains a list of names and phone numbers separated by spaces in the following format:</p> <ul style="list-style-type: none">a) Phone Number contains a 3- or 2-digit area code and a hyphen followed by an 8-digit number.b) Find all names having phone numbers with a 3-digit area code using regular expressions.

ASSIGNMENT-3

1.	Write a Python program to define a function called greet that takes a person's name as an argument and prints a greeting message. Demonstrate calling this function with different names.
2.	Write a Python program by creating a function ' calculate_area ' that calculates and returns the area of a rectangle. The function should take two parameters: length and width. Test the function with different values.
3.	Write a Python program to create a function power that raises a number to a specified power. Make the power parameter optional with a default value of 2.
4.	Write a Python program to define a function ' divide_and_remainder ' that takes two numbers as input and returns both the quotient and remainder when the first number is divided by the second.
5.	Write a Python program that demonstrates the difference between global and local variables inside and outside a function.
6.	Write a Python program to implement a recursive function to calculate the factorial of a non-negative integer.
7.	Write a Python program to create a Python decorator called timer that calculates and prints the execution time of a function. Apply this decorator to a sample function.
8.	Write a Python program that define two functions, square(x) and double(x) , that perform mathematical operations on a number. Use function composition to create a new function called ' square_and_double(x) ' that squares the number and then doubles the result.
9.	Write a Python program to create a function called average that calculates the average of any number of arguments passed to it. Test the function with different numbers of arguments.
10.	Write a function ' get_math_function(operation) ' that takes an operation (e.g., "add," "subtract") and returns the corresponding mathematical function. Use this function to perform operations on numbers.

ASSIGNMENT-4

1.	Write a Python program that takes two strings as input from the user and concatenates them to create a new string. Display the result.
2.	Write a Python program to create a function that calculates and returns the length of a given string without using the built-in len() function.
3.	Write a Python program that reverses a given string and displays the reversed string as the output.
4.	Given a string, write a Python program that extracts and displays the first three characters, the last three characters, and a substring from the middle of the string.
5.	Write a Python program that asks the user to enter a sentence and then displays the sentence in all uppercase and all lowercase without using any built-in function.
6.	<p>Write a Python program that takes a sentence as input from the user and performs the following transformations using string functions:</p> <ul style="list-style-type: none">a) Remove any leading or trailing whitespace from the input.b) Convert the sentence to lowercase.c) Replace all spaces with underscores ('_').d) Display the transformed sentence as the output. <p>For example, if the user enters " This is a Sample String ," the program should output "this_is_a_sample_string."</p>
7.	Write a Python program to create function called ' capitalize_words ' that takes a string as input and returns the same string with the first letter of each word capitalized. Use string functions to achieve this.
8.	Write a Python program that checks if a given string is a palindrome (reads the same forwards and backwards) and returns a boolean result.
9.	Write a Python program to create a function that takes a string as input and counts the frequency of each character (case-insensitive). Display the results in a dictionary.
10.	Write a Python program that removes all vowels from a given string and displays the modified string.

ASSIGNMENT-5

1.	Write a Python program to create a function that takes a list of numbers as input and calculates and returns the sum of all the numbers in the list.
2.	Write a Python program that concatenates two lists and displays the resulting list. For example, if you have lists [1, 2, 3] and [4, 5, 6], the program should return [1, 2, 3, 4, 5, 6].
3.	Write a Python function that takes a list and returns a new list with its elements reversed. Do not use the reverse() method.
4.	Given a list of numbers, write a python program that extracts and displays a portion of the list (a slice) based on user-defined start and end indices.
5.	Write a Python program to create a function that takes a list of numbers with duplicates and returns a new list with duplicates removed while preserving the original order.
6.	Write a Python program that sorts a list of strings in ascending order and displays the sorted list. Ensure the original list remains unchanged.
7.	Write a Python program that transposes a given 2D list (matrix) by converting its rows into columns and columns into rows.
8.	Develop a python program that takes two lists and returns a new list containing elements that are common between the two input lists.
9.	Write a Python program to create a function that rotates the elements of an array (list) to the right by a given number of positions.
10.	Write a Python program that finds and displays the largest and smallest elements in a list without using built-in functions.

ASSIGNMENT-6

1.	Write a Python program that creates a new text file named "sample.txt" and writes the text "Hello, World!" to it. Ensure that the file is properly closed after writing.
2.	Write a Python program that reads and displays the contents of the "sample.txt" file created in the previous question.
3.	Extend the previous program to append the text "Appending to the file" to the end of the "sample.txt" file without overwriting its existing content.
4.	Write a Python program that copies the content of one text file (e.g., "source.txt") to another text file (e.g., "destination.txt"). Ensure that the program works for files of different sizes.
5.	Develop a Python program that deletes the "sample.txt" file if it exists. Check if the file exists before attempting to delete it.
6.	Write a Python program that reads a text file and counts the number of words in it. Display the word count as the output.
7.	Write a Python program that reads a text file and counts the frequency of each character (letters, digits, symbols) in the file. Display the results in a dictionary.
8.	Write a Python program that reads the contents of a text file, reverses the order of lines, and writes the reversed lines to a new file.
9.	Write a Python program that reads a text file and replaces all occurrences of a specified word or phrase with another word or phrase. Save the modified text to a new file.
10.	Write a program that searches for a specific word or phrase in all text files within a specified directory and its subdirectories. Display the list of matching files.

ASSIGNMENT-7

1.	Write a Python program that asks the user to input two numbers and handles the "ZeroDivisionError" exception if the second number is zero. Display an error message in such cases.
2.	Write a Python program that attempts to open and read a file ("data.txt") and handles the "FileNotFoundError" exception if the file does not exist. Display an appropriate message.
3.	Implement a Python program that asks the user to input a number and handles the "ValueError" exception if the input cannot be converted to an integer. Ask the user to try again.
4.	Write a Python program that attempts to access an element in a list using an index that is out of range and handles the "IndexError" exception. Display an error message.
5.	Create a Python program that opens a file, reads its contents, and performs some operations on the data. Handle both "FileNotFoundError" and "IOError" exceptions and display relevant messages.
6.	Write a Python class called "Student" with attributes like "name" and "age." Create an instance of the class and display its attributes.
7.	Define a class "Rectangle" with methods to calculate the area and perimeter of a rectangle. Create an instance of the class and calculate these values.
8.	Create a class called "Person" with a constructor to initialize the name and age of a person. Display the person's details using an object of the class.
9.	Define a base class "Animal" with attributes like "name" and "species." Create a derived class "Dog" that inherits from "Animal" and adds a "breed" attribute. Display the attributes of a "Dog" object.
10.	Write a class "BankAccount" that uses encapsulation to protect the account balance. Implement methods for deposit, withdrawal, and displaying the balance.

ASSIGNMENT-8

1.	Create a Python module (e.g., math_operations.py) that contains functions for addition, subtraction, multiplication, and division. Write a program that imports this module and uses its functions to perform arithmetic operations.
2.	Write a Python program that imports the "math" module and calculates the square root of a number using the alias "m" for the module.
3.	Create a module with several functions, and then write a program that imports only a specific function from that module and uses it in python.
4.	Define a package with multiple submodules, and then write a program that imports a specific function from one of the submodules using python.
5.	Create a Python module that contains mathematical constants (e.g., pi, e). Write a program that imports the module and uses these constants to calculate the circumference of a circle.
6.	Create a custom Python package with at least two modules. Write a program that imports and uses functions or classes from different modules within the package.
7.	Create a python program with a variable named "x" in the global namespace and a module with a function that also uses a variable named "x." Demonstrate how the namespace conflict is resolved.
8.	Write a Python program that defines a variable in the global scope, and then defines a function with a local variable of the same name. Demonstrate how the local scope affects variable access.
9.	Create a Python package with an __init__.py file. Write a program that imports and uses a module from within the package.
10.	Write a Python program that uses the "requests" library to make an HTTP GET request to a URL and displays the response content.

ASSIGNMENT-9

1.	Write a Python program to print the current working directory then change the current working directory to a specified path using the os module.
2.	Create a program that deletes a file from a specified path using the os module.
3.	Write a Python program using the os module to list all files in a given directory and counts the number of files in that given directory.
4.	Create a program that deletes a file from a specified path using the os module.
5.	Create a program that calculates and prints the total size of all files in a specified directory (including subdirectories) using the os module.
6.	Implement a program that renames all files in a directory by adding a prefix or suffix to their names using the os module.
7.	Write a program that sorts all files in a directory based on their size and date modified using the os module.
8.	Create a program that lists all files with a specific extension in a given directory using the os module.

ASSIGNMENT-10

1.	Write a Python program to create a NumPy array from list, tuple with float type
2.	Write a Python program to demonstrate ndim, shape, size, dtype, slicing, integer and boolean array indexing.
3.	Write a Python program to find min, max, sum, cumulative sum, rank, determinant, and trace of array.
4.	Write a Python program to find eigenvalues and eigenvectors of a square matrix using NumPy.
5.	Write a Python program to find matrix and vector products (dot, inner, outer, product), matrix exponentiation.
6.	Write a Python program to solve a linear matrix equation, or system of linear scalar equations.
7.	Write a Python program to perform Sorting, Searching and Counting using NumPy methods.
8.	Write a Python program to demonstrate the use of the reshape() method.
9.	Write a Python program to concatenate two NumPy arrays along a specified axis
10.	Write a Python program to generate a NumPy array of size 5x5 with random integer values between 1 and 100, then invert it.
11.	Write a Python program to generate a NumPy array with datetime values representing timestamps for the last seven days.
12.	Write a Python program to Reshape a 1D NumPy array generated using random number into a 2D array of shape (5,5) and then transpose it.
13.	Write a Python program to create two NumPy arrays of the same size and perform element-wise addition, subtraction, multiplication, and division.

ASSIGNMENT-11

1.	Write a Python program to create a Pandas Series from a dictionary.
2.	Write a Python program that Create a NumPy array with random integer values representing temperatures for 100 different days. Convert this array into a Pandas DataFrame with columns 'Day' and 'Temperature' and store it in a excel sheet.
3.	Write a Python program which make use of following Pandas methods a. describe() b. head() c. tail()
4.	Write a Python program to get n-largest and n-smallest values from a particular column in Pandas DataFrame
5.	Write a Python program to create a Pandas DataFrame from a dictionary of lists containing student data (Name, Age, Grade).
6.	Write a Python program to merge two Pandas DataFrames based on a common key, such as a student ID, and display the combined DataFrame.
7.	Write a Python program to read a time-series data that are previously created and perform basic time series analysis, such as calculating rolling averages.
8.	Write a Python program to group a DataFrame by a specific column (e.g., Grade) and calculate the average age and maximum grade within each group.
9.	Write a Python program to calculate descriptive statistics (mean, median, standard deviation) for numeric columns in a Pandas DataFrame.
10.	Write a Python program that convert categorical data to numerical representation using Pandas for a DataFrame with a categorical column.
11.	Write a Python program to create a pivot table from a Pandas DataFrame containing sales data, summarizing the total sales for each product and region.
12.	Write a Python program using pandas that finds Missing Data and replace missing data.

ASSIGNMENT-12

1.	Write a Python program to create a scatter plot using Matplotlib and Seaborn to visualize the relationship between two numerical variables and add a regression line to the scatter plot using Seaborn.
2.	Write a Python program to read a dataset containing categorical variables using a Pandas DataFrame. Create a bar chart using Seaborn to show the count of each category. Use the 'hue' parameter in Seaborn to differentiate the bars by another categorical variable.
3.	Write a Python program to generate some random data, then create a box plot using Seaborn to show the distribution of each variable. Identify and mark the outliers on the box plot.
4.	Write a Python program to create a box plot using Seaborn to show the distribution of each variable. Identify and mark the outliers on the box plot.
5.	Write a Python program to load a dataset with multiple numerical variables into a Pandas DataFrame. Create a figure with multiple subplots using Matplotlib. Plot histograms for each variable in separate subplots.
6.	Write a Python program to create a line plot using Seaborn to visualize the trend over time. Customize the plot to include labels, titles, and any additional relevant information.
7.	Write a Python program to create a plot of your choice (scatter, line, bar, etc.) using Seaborn or Matplotlib. Apply customization and styling options (change colors, add titles, adjust axis labels) to make the plot visually appealing.