

SAGE UNIVERSITY

INDORE



Institute of Advance Computing

LAB MANUAL

B.Tech Practical

Academic Session: 2024-2025

Program : Bachelor of Technology in CST

Course Code : ACTDSBPY001P

Course Name : Basics of Python Programming

Submitted by :-

Name :- Sourabh Singh

Course Coordinator

Name: Prof. Gopal Goud

Enrollment no :-24ADV3CST0165

Signature :

SAGE University, Indore

Institute Name: Institute of Advance Computing		Department Name: Advance Computing						
Recommended Programs:					Semester: I			
Course Name Basics of Python Programming Lab		Course Code ACTDSBPY001P						
Credit Hours	L	T	P	N	Total Credits	2		
	0	0	4	0				
Prerequisites (if any)	Mathematics of 11 th and 12 th							
Course Objectives	In this course student will learn: <ul style="list-style-type: none"> • How to write simple programs using Python, a dynamic object-oriented programming language that can be used for software development projects. • To apply data structures and programming constructs in the Python language. • To apply the object-oriented programming concepts. • To demonstrate the use of exceptions handling, file handling. • To apply the concept of multithreading and multiprocessing in Python. 							

Course Content	<p>List of Experiments/Labs:</p> <ol style="list-style-type: none"> 1. Write a program in PYTHON to display a Message. 2. Write a program in PYTHON to take 3 numbers as input from User, compute Average and display the Result. 3. Write a program in PYTHON to take multiple inputs from User in a single line. 4. Write a program in PYTHON to check whether the input Number is 0 or 1 or Even or Odd. 5. Write a program in PYTHON to print Table of input Number using WHILE Loop. 6. Write a program in PYTHON to print the Pyramid of order “n” using FOR Loop <pre style="margin-left: 40px;">* * * * * * * * * * * * * * *</pre> 7. Write a program in PYTHON to perform STRING Operations. 8. Write a program in PYTHON to initialize a List, add “n” Elements in it, Sort the List and display the Sorted List. 9. Write a program in PYTHON to perform various Operations on List.
-----------------------	---

	<p>10. Write a program in PYTHON to implement Linear Search on List Elements.</p> <p>11. Write a program in PYTHON to Sort a List using Selection Sort Algorithm.</p> <p>12. Write a program in PYTHON to implement a User defined Function.</p> <p>13. Write a program in PYTHON to implement Array using NUMPY.</p> <p>14. Write a program in PYTHON to perform Matrix Addition and Multiplication.</p> <p>15. Write a program in PYTHON to perform Traversing Operations on 2 D Array.</p> <p>16. Write a program in PYTHON to demonstrate the use of Mathematical Functions defined in NUMPY.</p> <p>17. Write a program in PYTHON to draw Shapes / Graphics using TURTLE.</p> <p>18. Write a program in PYTHON to plot a Graph using MATPLOTLIB.</p> <p>19. Write a program in PYTHON to implement File Handling.</p> <p>20. Write a program in PYTHON to implement Exception Handling.</p>
Text Books	T1: Starting Out with Python, Toney Gaddis, Third Edition Pearson.
References	R1: Python Data Science Handbook, Jake Vanderplas,

Course Outcomes	<p>After completion of this course, students are able to:</p> <ul style="list-style-type: none"> CO1. Comprehend the basics of Python variable, data types, I/O functions, selection, and iteration. CO2. Apply data structures and programming constructs in the Python language. CO3. Apply the object-oriented programming concepts. CO4. Demonstrate the use of exceptions handling, file handling. CO 5. Apply the concept of multithreading and multiprocessing in Python.
------------------------	--

Mapping of Course outcome with Program Outcomes, PSO's, and Knowledge Levels (As per Blooms Taxonomy)

CO/P O	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	PSO 1	PSO 2	PSO 3	Knowledg e Levels (K ₁ , K ₂ , ..., K ₆)	
CO1	3	2	1			3					2			1	1		K1, K2
CO2	1	2	3			1							2		1		K1, K2, K3
CO3	1	2	3	2		1					2			1			K1, K2, K3,K4, K5, K6

CO4			2	1	3						1			2	K1, K2, K3, K4
CO5	2	1	2	3	3	2					1	2			KI, K2, K3

High-3

Medium-2

Low-1

K₁=>Remember K₂=>Understand K₃=>Apply K₄=>Analyze K₅=>Evaluate

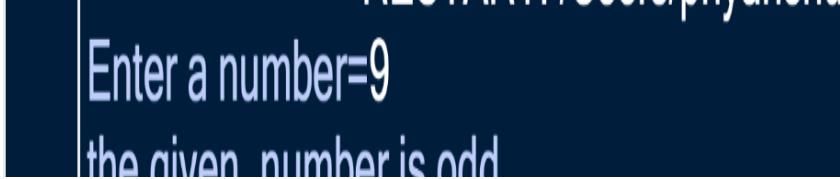
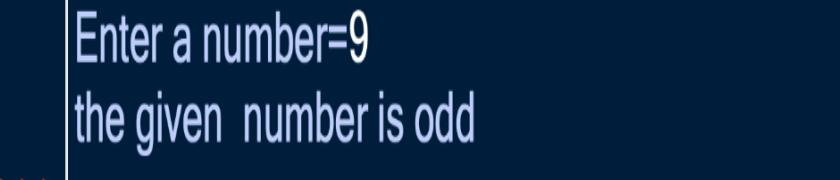
K₆=>Create

Designed By: Prof. Neha Sharma (Name with Sign.)	Checked By: Prof. Anand Jawdekar (Name with Sign.)	Approved By: Dr. Manoj Ramaiya (Name with Sign.)
--	--	--

Experiment Number	1
Experiment Title	Write a program in PYTHON to display a Message.
Course Outcome (CO)	C01. How to write simple programs using python, a dynamic object-oriented programming language that can be used for software development projects
Apparatus/softwares Required	<p>:- A computer with Python installed (Python 3 x version recommended)</p> <p>:- A text editor or Integrated Development Environment (IDE) such as IDLE, VS Code, or PyCharm</p>
Experiment Description	The aim of this experiment is to create a basic python program that displays a predefined message on the screen. This will help students understand how to write simple code , use the print () function, and run a Python program.
Experiment Code/Procedure	<pre># Program to display a message print ("Hello, World!")</pre>
Input	N/A
Output/Result	Hello, World!

Experiment Number	2
Experiment Title	Write a program in PYTHON to take 3 numbers as input from User, compute Average and display the Result.
Course Outcome (CO)	C01.How to take input from user of 3 numbers and print their average using python that can be used for software development projects.
Tools/Apparatus Required	<p>:- A computer with Python installed (Python 3 x version recommended)</p> <p>:- A text editor or Integrated Development Environment (IDE) such as IDLE, VS Code, or PyCharm</p>
Experiment Description	The aim of this experiment is to create a basic python program that takes 3 numbers as input and prints their average. This will help us to understand how to find the average of numbers ,use the print() Function, and run a python program.
Experiment Code/Procedure	<pre>#program to find the average of 3 numbers a=int(input("Enter the first number=")) b=int(input("Enter the second number=")) c=int(input("Enter the third number=")) sum=a+b+c; average=sum/3 print(average)</pre>
Input	<pre>Enter the first number=8 Enter the second number=9 Enter the third number=7 8.0</pre>
Output/Result	<pre>Enter the first number=8 Enter the second number=9 Enter the third number=7 8.0 >>></pre>

Experiment Number	3
Experiment Title	Write a program in PYTHON to take multiple inputs from User in a single line.
Course Outcome (CO)	How to take multiple inputs in a single line using python, a dynamic object oriented programming language that can be used for software development.
Tools/Apparatus Required	A computer with python installed(Python 3.13version recommended) A text editor or integrated development environment(IDE) such as IDLE,VS code.
Experiment Description	The aim of this experiment is to create a basic python program that takes multiple input from user in a single line.Using print() function, and Split() function to run a python program.
Experiment Code/Procedure	<pre># program to display 3 inputs from user in a single line x, y, z = input("Enter three values: ").split() print("Total number of students: ", x) print("Number of passed student : ", y) print("Number of failed student : ", z) print()</pre>
Input	<p>Enter three values: 89 67 56</p> <p>Total number of students: 89</p>
Output/Result	<p>Enter three values: 89 67 56</p> <p>Total number of students: 89</p> <p>Number of passed student : 67</p> <p>Number of failed student : 56</p>

Experiment Number	4
Experiment Title	Write a program in PYTHON to check whether the input Number is 0 or 1 or Even or Odd.
Course Outcome (CO)	How to check that the input of the numbers is 0,1,even or odd using python program, a dynamic object oriented programming language that can be used for software development projects.
Tools/Apparatus Required	.A computer with python installed(Python3.13 version recommended) .A text editor or integrated development environment(IDE) such as IDLE,VS Code.
Experiment Description	The aim of this experiment is to create a basic python program that can tell the user whether the given number is 0 or 1 or even or odd. Using if else statement and print() function.This will help us to understand the concept of if else experiment.
Experiment Code/Procedure	<pre>#program to check whether the given condition is even or odd or 1 or 0. num=int(input("Enter a number=")) if (num%2==0): print("the given number is even") else: print("the given number is odd")</pre>
Input	
Output/Result	

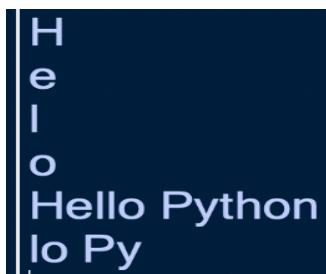
Experiment Number	5
Experiment Title	Write a program in PYTHON to print table of input number using while loop
Course Outcome (CO)	How to print table of input number using python program, a dynamic object oriented programming language that can be used for software development projects in this we learn the concept of while loop
Tools/Apparatus Required	A computer with python installed(Python3.13 version recommended). A text editor or integrated development environment(IDE) such as IDLE,VS Code.
Experiment Description	The aim of this experiment is to create a basic python program that can tell the user table of input number using while loop and print() function. This will help us to understand the concept of while, for loop
Experiment Code/Procedure	#program to print a table using while loop. n=int(input("Enter a number=")) i=1 While i<11: value=n*1 print(n,"*",i,"=",value) i=i+1
Input	Enter any Number :8 8 * 1 = 8
Output/Result	Enter any Number :8 8 * 1 = 8 8 * 2 = 16 8 * 3 = 24 8 * 4 = 32 8 * 5 = 40 8 * 6 = 48 8 * 7 = 56 8 * 8 = 64 8 * 9 = 72 8 * 10 = 80

Experiment Number	6
Experiment Title	Write a program in PYTHON to print the pyramid of order “n” using while loop
Course Outcome (CO)	How to print table of pyramid of order “n” using python program, a dynamic object oriented programming language that can be used for software development projects in this we learn the concept of while loop
Tools/Apparatus Required	.A computer with python installed(Python3.13 version recommended) .A text editor or integrated development environment(IDE) such as IDLE,VS Code.
Experiment Description	The aim of this experiment is to create a basic python program that can show the user pyramid of order “n” using while loop and print() function. This will help us to understand the concept of while, for loop
Experiment Code/Procedure	<pre># Function to print full pyramid pattern def full_pyramid(n): for i in range(1, n + 1): # Print leading spaces for j in range(n - i): print(" ", end="") # Print asterisks for the current row for k in range(1, 2*i): print("*", end="") print() full_pyramid(5)</pre>

Input	
Output/Result	<pre>● radarr@radarr: ~ % p * *** **** *****</pre>

Experiment Number	7
Experiment Title	Write a program in PYTHON to perform STRING operation
Course Outcome (CO)	How to perform STRING operation using python program, a dynamic object oriented programming language that can be used for software development projects in this we learn the concept of for loop
Tools/Apparatus Required	.A computer with python installed(Python3.13 version recommended) .A text editor or integrated development environment(IDE) such as IDLE,VS Code.
Experiment Description	The aim of this experiment is to create a basic python program that can show the user string operation using for loop and print() function. This will help to understand concept of for loop
Experiment Code/Procedure	#program to perform string operations. str="Hello Python" print(str[0]) print(str[1]) print(str[2]) print(str[4]) print(str[3:8])
Input	

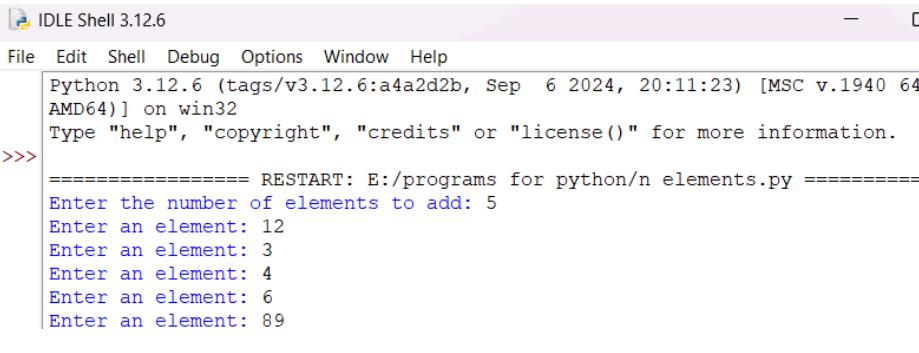
Output/Result



A screenshot of a terminal window with a dark blue background and white text. The text displays the word "Hello" on four separate lines, followed by "Hello Python" on one line, and "lo Py" on two lines below it. This output is likely from a Python script that prints multiple lines of text.

```
Hello  
Hello  
Hello  
Hello Python  
lo Py
```

Experiment Number	8
Experiment Title	Write a program in PYTHON to initialize a list add “n” Element in it, sort the List and display the sorted list
Course Outcome (CO)	How to perform a program to initialize a list add “n” Element in it, sort the List and display the sorted list using python program, a dynamic object oriented programming language that can be used for software development projects in this we learn the concept of for loop
Tools/Apparatus Required	
Experiment Description	The aim of this experiment is to create a basic python program that can show the user sort the list and display the sorted list using print() function. This will help to understand concept of a list in python.
Experiment Code/Procedure	<pre># Initialize an empty list my_list = [] n = int(input("Enter the number of elements to add: ")) for _ in range(n): element = int(input("Enter an element: ")) my_list.append(element) my_list.sort() print("Sorted List:", my_list)</pre>

Input	 <pre>IDLE Shell 3.12.6 File Edit Shell Debug Options Window Help Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep 6 2024, 20:11:23) [MSC v.1940 64 AMD64] on win32 Type "help", "copyright", "credits" or "license()" for more information. >>> ===== RESTART: E:/programs for python/n elements.py ===== Enter the number of elements to add: 5 Enter an element: 12 Enter an element: 3 Enter an element: 4 Enter an element: 6 Enter an element: 89</pre>
Output/Result	Sorted List: [3, 4, 6, 12, 89]

Experiment Number	9
Experiment Title	Write a program in PYTHON to perform various Operations on List.
Course Outcome (CO)	How to perform various operations on list using python, a dynamic object oriented programming language that can be used for software development projects.
Tools/Apparatus Required	
Experiment Description	The aim of this experiment is to perform various operations on list
Experiment Code/Procedure	<pre># Initialize an empty list my_list = [] n = int(input("Enter the number of elements to add: ")) for _ in range(n): element = int(input("Enter an element: ")) my_list.append(element) print("Original List:", my_list) my_list.sort() print("Sorted List:", my_list) my_list.reverse() print("Reversed List:", my_list) new_element = int(input("Enter a new element to add: "))</pre>

	<pre> my_list.append(new_element) print("List after adding new element:", my_list) remove_element = int(input("Enter an element to remove: ")) if remove_element in my_list: my_list.remove(remove_element) print("List after removing the element:", my_list) else: print("Element not found in the list.") print("Length of the List:", len(my_list)) </pre>
Input	 IDLE Shell 3.12.6 File Edit Shell Debug Options Window Help Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep 15 2023, 16:46:46) [GCC 11.3.0] AMD64] on win32 Type "help", "copyright", "credits" or >>> = RESTART: E:/C++ programs/list.py Enter the number of elements to add: 5 Enter an element: 34 Enter an element: 45 Enter an element: 78 Enter an element: 98 Enter an element: 345
Output/Result	Original List: [34, 45, 78, 98, 345] Sorted List: [34, 45, 78, 98, 345] Reversed List: [345, 98, 78, 45, 34] Enter a new element to add: 4567 List after adding new element: [345, 98, 78, 45, 34, 4567] Enter an element to remove: 45 List after removing the element: [345, 98, 78, 34, 4567] Length of the List: 5

Experiment Number	10
Experiment Title	Write a program in PYTHON to implement Linear Search on List Elements.
Course Outcome (CO)	How to implement linear search on list elements, using dynamic object oriented programming language that can be used for software development projects.
Tools/Apparatus Required	A computer with python installed (Python 3.13 version recommended) . A text editor or integrated development environment (IDE)such as IDLE, VS code.
Experiment Description	The aim of this experiment is to perform linear search on list elements
Experiment Code/Procedure	<pre># Initialize a list my_list = [] n = int(input("Enter the number of elements: ")) for _ in range(n): element = int(input("Enter an element: ")) my_list.append(element) key = int(input("Enter the element to search for: ")) found = False</pre>

```

for i in range(len(my_list)):

    if my_list[i] == key:

        print(f"Element {key} found at index {i}.") 

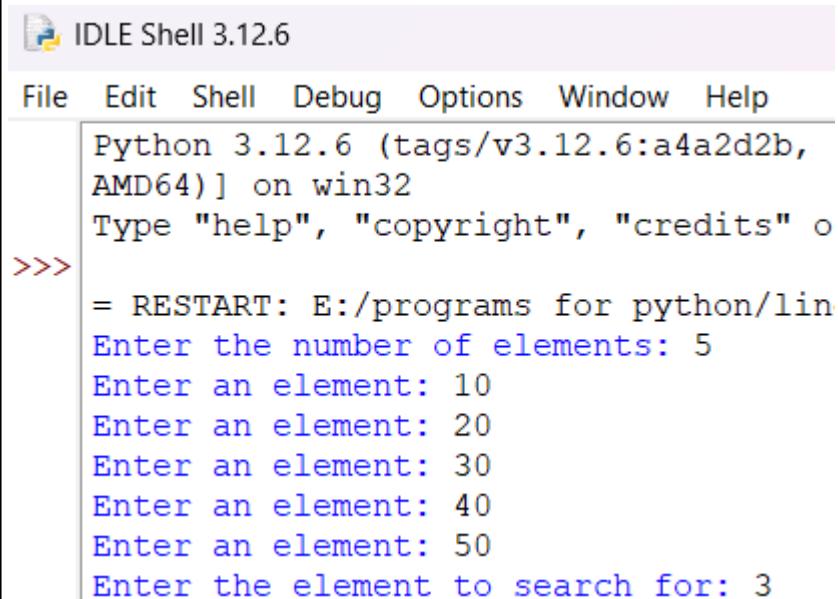
        found = True

        break

if not found:

    print(f"Element {key} not found in the list.")

```

Input


The screenshot shows the IDLE Shell 3.12.6 interface. The title bar says "IDLE Shell 3.12.6". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays a Python session:

```

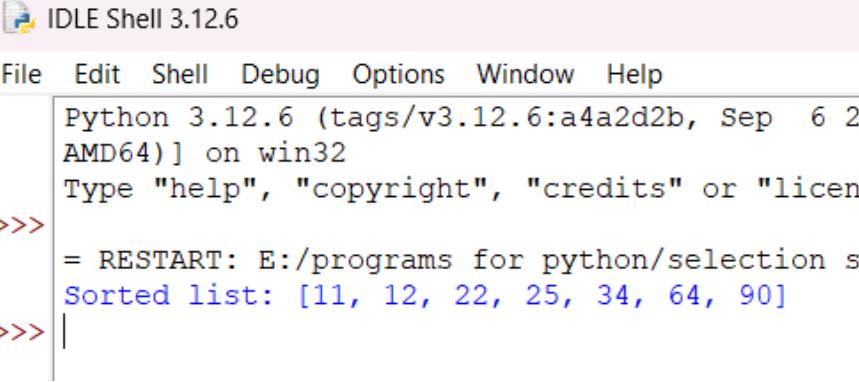
Python 3.12.6 (tags/v3.12.6:a4a2d2b, AMD64) ] on win32
Type "help", "copyright", "credits" o
>>>
= RESTART: E:/programs for python/list.py
Enter the number of elements: 5
Enter an element: 10
Enter an element: 20
Enter an element: 30
Enter an element: 40
Enter an element: 50
Enter the element to search for: 3

```

Output/Result

Element 3 not found in the list.

Experiment Number	11
Experiment Title	Write a program in PYTHON to Sort a List using Selection Sort Algorithm.
Course Outcome (CO)	How to implement the Selection Sort algorithm on a list using Python, a dynamic object-oriented programming language widely used for software development projects
Tools/Apparatus Required	A computer with python installed (Python 3.13 version recommended) . A text editor or integrated development environment (IDE)such as IDLE, VS code.
Experiment Description	The aim of this experiment is to sort a list using selection sort
Experiment Code/Procedure	<pre># Simple Selection Sort Algorithm data = [64, 34, 25, 12, 22, 11, 90] for i in range(len(data)): min_index = i for j in range(i + 1, len(data)): if data[j] < data[min_index]: min_index = j data[i], data[min_index] = data[min_index], data[i]</pre>

	print("Sorted list:", data)
Input	N/A
Output/Result	 <p>IDLE Shell 3.12.6</p> <p>File Edit Shell Debug Options Window Help</p> <p>Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep 6 2023, 16:45:49) [MSC v.1932 64 bit (AMD64)] on win32</p> <p>Type "help", "copyright", "credits" or "license" for more information.</p> <pre>>>> = RESTART: E:/programs for python/selection sort.py Sorted list: [11, 12, 22, 25, 34, 64, 90] >>></pre>

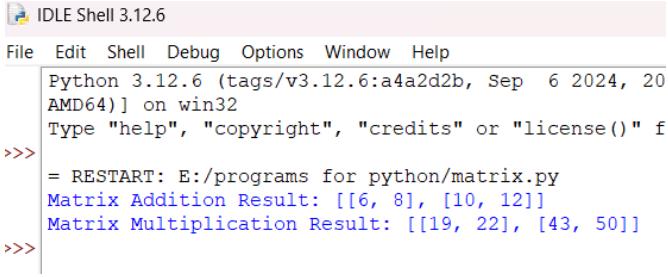
Experiment Number	12
Experiment Title	Write a program in PYTHON to implement a User defined Function.
Course Outcome (CO)	How to create and use user-defined functions in Python, a versatile object-oriented programming language, to enhance code modularity and reusability for software development projects.
Tools/Apparatus Required	A computer with python installed (Python 3.13 version recommended) . A text editor or integrated development environment (IDE)such as IDLE, VS code.
Experiment Description	The aim of this experiment is to implement a user-defined function that calculates the square of a number.
Experiment Code/Procedure	<pre># User-defined function to calculate the square of a number def calculate_square(number): return number * number num = int(input("Enter a number: ")) result = calculate_square(num) print("The square of", num, "is:", result)</pre>

Input	 IDLE Shell 3.12.6 File Edit Shell Debug Opt Python 3.12.6 (tags AMD64)] on win32 Type "help", "copyright" >>> = RESTART: E:/progr Enter a number: 5
Output/Result	 IDLE Shell 3.12.6 File Edit Shell Debug Options Python 3.12.6 (tags/v3. AMD64)] on win32 Type "help", "copyright" >>> = RESTART: E:/programs Enter a number: 5 The square of 5 is: 25

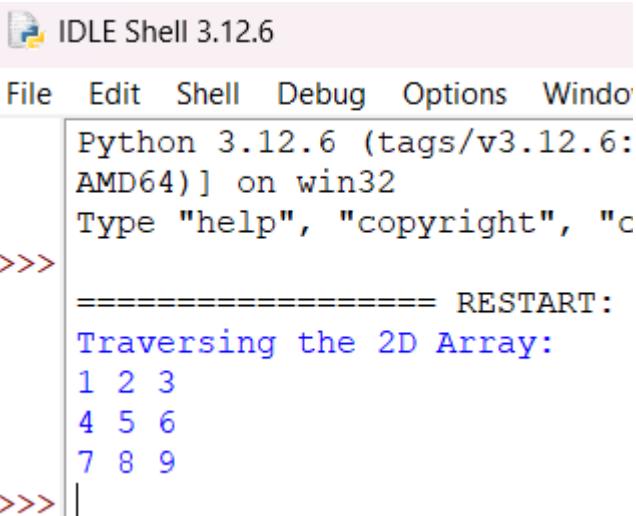
Experiment Number	13
Experiment Title	Write a program in PYTHON to implement Array using NUMPY.
Course Outcome (CO)	How to implement and manipulate arrays using NumPy in Python, a powerful library for numerical computations, to enhance efficiency in handling large datasets and performing mathematical operations in software development projects
Tools/Apparatus Required	A computer with python installed (Python 3.13 version recommended) . A text editor or integrated development environment (IDE)such as IDLE, VS code.
Experiment Description	The aim of this experiment is to implement array using NUMPY and print sum and mean value of elements
Experiment Code/Procedure	<pre>import numpy as np arr = np.array([1, 2, 3, 4, 5]) print("NumPy Array:", arr) print("Sum of elements:", np.sum(arr)) print("Mean of elements:", np.mean(arr))</pre>

Input	N/A
Output/Result	<pre>NumPy Array: [1 2 3 4 5] Sum of elements: 15 Mean of elements: 3.0</pre>

Experiment Number	14
Experiment Title	Write a program in PYTHON to perform Matrix Addition and Multiplication.
Course Outcome (CO)	How to perform matrix addition and multiplication in Python, using functions and list comprehensions, to enhance understanding of matrix operations and Python's capabilities for handling numerical data.
Tools/Apparatus Required	A computer with python installed (Python 3.13 version recommended) A text editor or integrated development environment (IDE)such as IDLE, VS code.
Experiment Description	The aim of this program is to perform matrix addition and multiplication using functions in Python. It demonstrates how to manipulate matrices and display results using list comprehensions and the <code>print()</code> function
Experiment Code/Procedure	<pre># Matrix Addition and Multiplication def add_matrices(matrix1, matrix2): return [[matrix1[i][j] + matrix2[i][j] for j in range(len(matrix1[0]))] for i in range(len(matrix1))] def multiply_matrices(matrix1, matrix2):</pre>

	<pre> return [[sum(matrix1[i][k] * matrix2[k][j] for k in range(len(matrix2))) for j in range(len(matrix2[0]))] for i in range(len(matrix1))] matrix1 = [[1, 2], [3, 4]] matrix2 = [[5, 6], [7, 8]] added_matrix = add_matrices(matrix1, matrix2) multiplied_matrix = multiply_matrices(matrix1, matrix2) print("Matrix Addition Result:", added_matrix) print("Matrix Multiplication Result:", multiplied_matrix) </pre>
Input	
Output/Result	 <pre> IDLE Shell 3.12.6 File Edit Shell Debug Options Window Help Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep 6 2024, 20 AMD64)] on win32 Type "help", "copyright", "credits" or "license()" f >>> = RESTART: E:/programs for python/matrix.py Matrix Addition Result: [[6, 8], [10, 12]] Matrix Multiplication Result: [[19, 22], [43, 50]] >>> </pre>

Experiment Number	15
Experiment Title	Write a program in PYTHON to perform Traversing Operations on 2 D Array.
Course Outcome (CO)	How to traverse and access elements in a 2D array using nested loops in Python, enhancing understanding of array manipulation and data traversal in software development projects
Tools/Apparatus Required	A computer with python installed (Python 3.13 version recommended) . A text editor or integrated development environment (IDE)such as IDLE, VS code.
Experiment Description	The aim of this program is to demonstrate how to traverse a 2D array in Python. It iterates through each row and element of the array, printing the elements using nested loops. This helps in understanding how to access and display data stored in a 2D array in Python.
Experiment Code/Procedure	<pre># 2D Array array = [[1, 2, 3], [4, 5, 6], [7, 8, 9]</pre>

	<pre>] # Traversing the 2D Array print("Traversing the 2D Array:") for row in array: for element in row: print(element, end=" ") print() # Newline</pre>
Input	
Output/Result	 <p>IDLE Shell 3.12.6</p> <p>File Edit Shell Debug Options Window</p> <p>Python 3.12.6 (tags/v3.12.6: AMD64)] on win32</p> <p>Type "help", "copyright", "c</p> <p>>>> ===== RESTART:</p> <p>Traversing the 2D Array:</p> <p>1 2 3 4 5 6 7 8 9</p> <p>>>> </p>

Experiment Number	16
Experiment Title	Write a program in PYTHON to demonstrate the use of Mathematical Functions defined in NUMPY.
Course Outcome (CO)	How to use NumPy's mathematical functions to perform various numerical operations such as sum, mean, standard deviation, square root, and exponential, enabling efficient data manipulation and analysis in Python.
Tools/Apparatus Required	A computer with python installed (Python 3.13 version recommended) . A text editor or integrated development environment (IDE)such as IDLE, VS code.
Experiment Description	The aim of this program is to demonstrate the use of NumPy's mathematical functions for performing operations like sum, mean, standard deviation, square root, and exponential on a NumPy array. It helps in understanding how to efficiently perform mathematical computations on large datasets, which is essential for data analysis, scientific computing, and numerical problem-solving in Python.
Experiment Code/Procedure	<pre>import numpy as np arr = np.array([1, 2, 3, 4, 5]) sum_arr = np.sum(arr)</pre>

	<pre> mean_arr = np.mean(arr) std_arr = np.std(arr) max_arr = np.max(arr) min_arr = np.min(arr) sqrt_arr = np.sqrt(arr) exp_arr = np.exp(arr) print("Array:", arr) print("Sum:", sum_arr) print("Mean:", mean_arr) print("Standard Deviation:", std_arr) print("Maximum:", max_arr) print("Minimum:", min_arr) print("Square Root:", sqrt_arr) print("Exponential:", exp_arr) </pre>
Input	
Output/Result	<pre> Squared: [1 4 9 16 25] Square Root: [1. 1.41421356 1.73205081 2. 2.23606798] Exponential: [2.71828183 7.3890561 20.08553692 54.59815003 148.4131591] Logarithm: [0. 0.69314718 1.09861229 1.38629436 1.60943791] Sine: [0.84147098 0.90929743 0.14112001 -0.7568025 -0.95892427] Cosine: [0.54030231 -0.41614684 -0.9899925 -0.65364362 0.28366219] Absolute: [1 2 3 4 5] </pre>

Experiment Number	17
Experiment Title	Write a program in PYTHON to draw Shapes / Graphics using TURTLE.
Course Outcome (CO)	How to use the Turtle module in Python to create and manipulate basic shapes and graphics, enhancing understanding of graphical programming and visual representation of concepts in software development projects.
Tools/Apparatus Required	A computer with python installed (Python 3.13 version recommended) . A text editor or integrated development environment (IDE)such as IDLE, VS code.
Experiment Description	The aim of this program is to demonstrate how to use the Turtle module in Python to draw shapes such as a square, triangle, and circle. It utilizes the turtle graphics library to control a turtle object that moves around the screen, drawing the shapes. This helps to understand basic graphics programming and how to control drawing operations in Python.
Experiment Code/Procedure	<pre>import turtle screen = turtle.Screen() screen.bgcolor("white")</pre>

```
# Create a turtle object

pen = turtle.Turtle()

pen.shape("turtle")

pen.color("blue")

# Draw a square

pen.penup()

pen.goto(-100, 100)

pen.pendown()

for _ in range(4):

    pen.forward(100)

    pen.right(90)

# Draw a triangle

pen.penup()

pen.goto(100, 100)

pen.pendown()

for _ in range(3):

    pen.forward(100)

    pen.right(120)

# Draw a circle

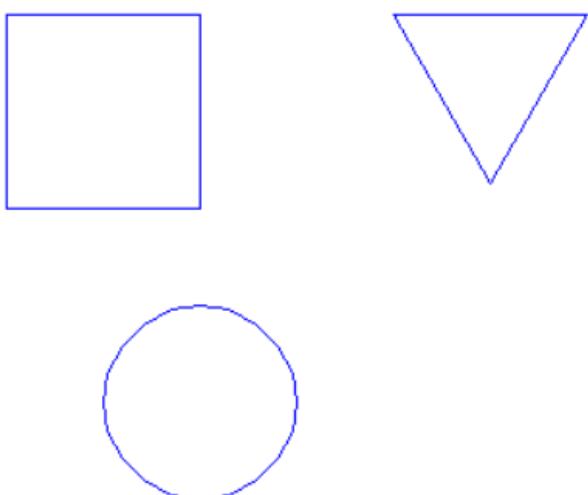
pen.penup()

pen.goto(0, -150)

pen.pendown()

pen.circle(50)

pen.hideturtle()
```

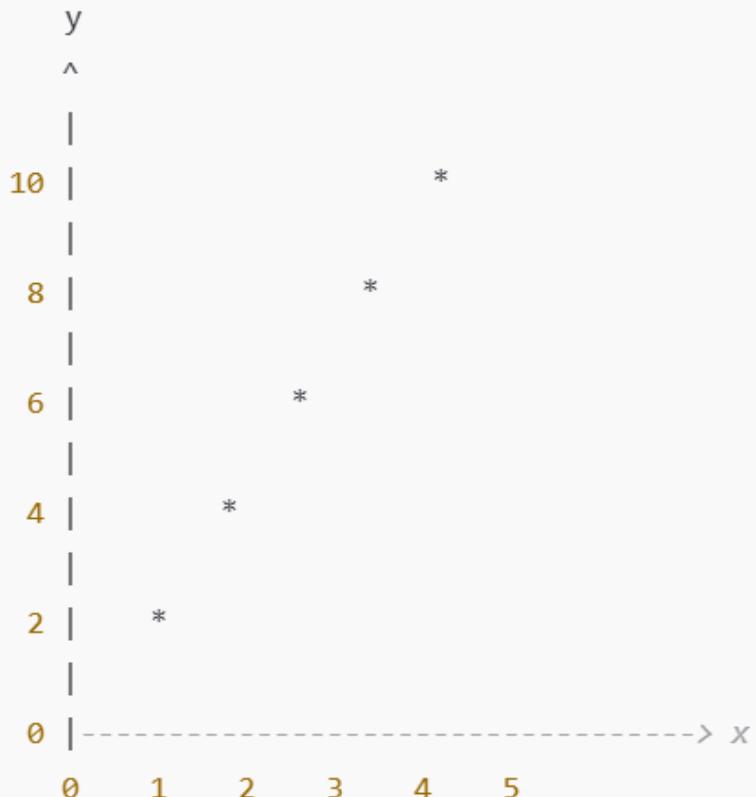
	turtle.done()
Input	N/A
Output/Result	 A blue-outlined rectangle, a blue-outlined triangle, and a blue-outlined circle.

Experiment Number	18
Experiment Title	Write a program in PYTHON to plot a Graph using MATPLOTLIB.
Course Outcome (CO)	How to use Matplotlib in Python to create and customize plots, enabling effective data visualization and graphical representation of data for software development and analysis
Tools/Apparatus Required	A computer with python installed (Python 3.13 version recommended) . A text editor or integrated development environment (IDE)such as IDLE, VS code.
Experiment Description	The aim of this experiment is to demonstrate how to use Matplotlib in Python to create a simple line plot. It covers the basic steps of plotting data, adding titles, labels, legends, and grids, as well as displaying the graph. This helps to understand the fundamentals of data visualization, which is essential for representing data in a meaningful and insightful way.
Experiment Code/Procedure	<pre>import matplotlib.pyplot as plt x = [0, 1, 2, 3, 4, 5]</pre>

```
y = [0, 1, 4, 9, 16, 25]  
plt.plot(x, y)  
plt.title('Simple Plot')  
plt.xlabel('X values')  
plt.ylabel('Y values')  
plt.show()
```

Input

Output/Result

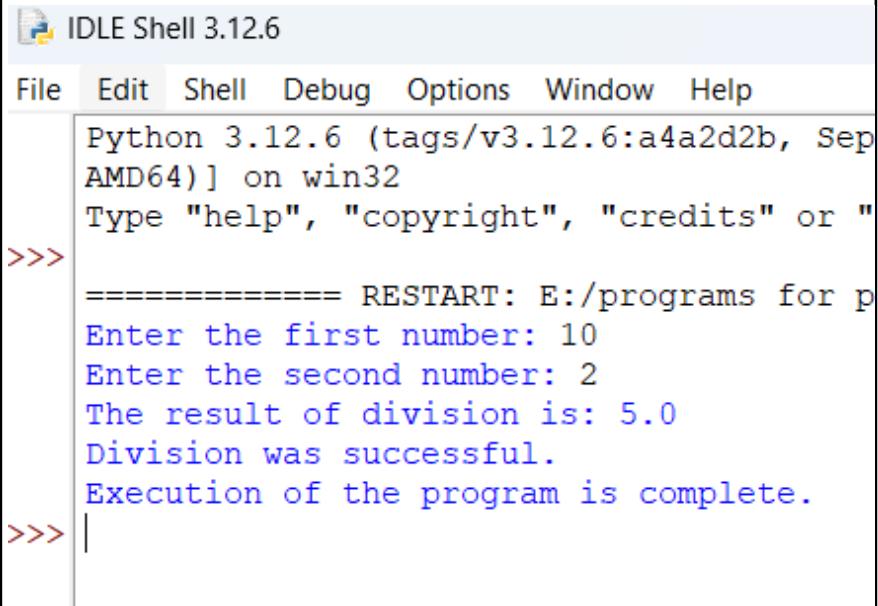


Experiment Number	19
Experiment Title	Write a program in PYTHON to implement File Handling.
Course Outcome (CO)	How to perform file handling operations in Python, including reading from and writing to files, to enhance skills in data storage and manipulation for software development projects.
Tools/Apparatus Required	A computer with python installed (Python 3.13 version recommended) . A text editor or integrated development environment (IDE)such as IDLE, VS code.
Experiment Description	The aim of this program is to demonstrate basic file handling operations in Python, such as creating, writing to, reading from, and closing a file. It helps understand how to work with files in Python, allowing users to store and retrieve data efficiently, which is essential for various real-world applications.
Experiment Code/Procedure	# Open a file in write mode and create the file if it doesn't exist file = open("example.txt", "w")

	<pre> file.write("Hello, this is a simple file handling example in Python.\n") file.write("Python makes file handling easy!\n") file.close() file = open("example.txt", "r") content = file.read() print("File Content:") print(content) file.close() </pre>
Input	N/A
Output/Result	 <p>IDLE Shell 3.12.6</p> <p>File Edit Shell Debug Options Window Help</p> <p>Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep 6 2024, 20:11:23) [GCC 11.2.0] on win32</p> <p>Type "help", "copyright", "credits" or "license()" for more</p> <p>>>> ===== RESTART: E:/programs for python/file handl</p> <p>File Content:</p> <p>Hello, this is a simple file handling example in Python. Python makes file handling easy!</p>

Experiment Number	20
Experiment Title	Write a program in PYTHON to implement Exception Handling.
Course Outcome (CO)	How to implement exception handling in Python to manage errors effectively using <code>try</code> , <code>except</code> , <code>else</code> , and <code>finally</code> blocks, enhancing the robustness and reliability of software applications.
Tools/Apparatus Required	A computer with python installed (Python 3.13 version recommended) . A text editor or integrated development environment (IDE)such as IDLE, VS code.
Experiment Description	The aim of this program is to demonstrate exception handling in Python, which helps manage runtime errors and ensures the program continues running smoothly. It uses <code>try</code> , <code>except</code> , <code>else</code> , and <code>finally</code> blocks to handle errors like division by zero and invalid input, providing a way to gracefully manage unexpected situations and prevent the program from crashing. This improves the robustness and user experience of Python applications.

Experiment Code/Procedure	<pre># Program to demonstrate exception handling try: num1 = int(input("Enter the first number: ")) num2 = int(input("Enter the second number: ")) result = num1 / num2 print("The result of division is:", result) except ZeroDivisionError: print("Error: Cannot divide by zero.") except ValueError: print("Error: Invalid input. Please enter integers.") except Exception as e: print("An unexpected error occurred:", e) else: print("Division was successful.") finally: print("Execution of the program is complete.")</pre>
Input	N/A

Output/Result

IDLE Shell 3.12.6

File Edit Shell Debug Options Window Help

Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep
AMD64)] on win32

Type "help", "copyright", "credits" or "

>>>

===== RESTART: E:/programs for p

Enter the first number: 10

Enter the second number: 2

The result of division is: 5.0

Division was successful.

Execution of the program is complete.

>>> |