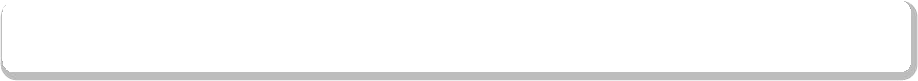
**GURU NANAK KHALSA   
COLLEGE**

**AUTONOMOUS (UNIVERSITY OF MUMBAI)**

**MUMBAI-400 019**

**DEPARTMENT OF COMPUTER SCIENCE**

**CERTIFICATE**

**A blue symbol with white text

AI-generated content may be incorrect.CERTIFIED that the practical’s, assignments duly signed, were performed by Mr/Ms. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Roll No: \_\_\_\_\_\_\_\_\_\_\_**

**of MSC.Big Data Analytics Part I for the Computer Science Laboratory of**

**G. N. Khalsa College ,Mumbai during the academic year 2025-2026.**

**He/ She has completed the course of Laboratory assignments in Computer Science as contained in the course prescribed by the University**

**of Mumbai.**

**Sign of Student Head Dept. of Computer Science**

**Date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Professor-in-charge Sign of Examiner’s**

**1] \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 1] \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**2] \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2] \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR.No** | **Practical Name** | **Date** | **Page No.** | **Sign** |
|  |  |  |  |  |
|  | Implement a program to determine the measurement scale (Nominal,Ordinal,Interval,Ratio) of a dataset. | 24/07/25 | 02 |  |
|  | Implement a program that calculates and interprets the skewness and kurtosis of a dataset. | 07/08/25 | 04 |  |
|  | Implement Eigenvalues and EigenVectors | 14/08/25 | 06 |  |
|  | Basics of Statistics:Frequency Table | 21/08/25 | 07 |  |
|  | Implement Object-Oriented Programming in Java A) Data Type B) Simple Calculator | 04/09/25 | 08 |  |
|  | Implement Bubble Sort in Java | 11/09/25 | 11 |  |
|  | Implement Selection Sort in Java | 18/09/25 | 12 |  |

**Practical 1**

**Aim: Implement a program to determine the measurement scale (Nominal, Ordinal, Interval, Ratio) of a dataset.**

**Program:**def detect\_scale(data):

    try:

        # Try converting all values to float

        numeric\_data = [float(x) for x in data]

        is\_numeric = True

    except:

        is\_numeric = False

    if not is\_numeric:

        return "Nominal"

    unique\_vals = sorted(set(numeric\_data))

    # Ratio scale: all values >= 0 and has a meaningful zero

    if min(numeric\_data) == 0 or all(x >= 0 for x in numeric\_data):

        if len(unique\_vals) > 10:

            return "Ratio"

        else:

            return "Ordinal"

    # Interval scale: contains negative values, but still numeric

    if min(numeric\_data) < 0:

        return "Interval"

    # Fallback

    return "Ordinal"

user\_input = input("Enter values separated by commas: ")

data = user\_input.split(",")  # split into list

result = detect\_scale(data)

print("Detected scale:", result)

**Output:**

* Nominal :  
  
* Ordinal:  
  
* Interval:  
  
* Ratio:  
  

**Practical 02**

**Aim: Implement a program that calculates and interprets the skewness and kurtosis of a dataset.**

**Program:**

from scipy.stats import skew, kurtosis

# Step 1: Take input from user

raw\_data = input("Enter ungrouped numerical data separated by commas: ")

# Step 2: Convert input string to a list of floats

try:

    data = [float(x.strip()) for x in raw\_data.split(",")]

except ValueError:

    print("Invalid input! Please enter only numbers separated by commas.")

    exit()

# Step 3: Calculate skewness and kurtosis

data\_skewness = skew(data)

data\_kurtosis = kurtosis(data)  # By default, Fisher's definition (normal dist = 0)

# Step 4: Display results

print("\n--- Results ---")

print("Ungrouped Data:", data)

# Skewness

print(f"\nSkewness: {data\_skewness:.4f}")

if data\_skewness > 0:

    print("Interpretation: Positively skewed (tail on the right)")

elif data\_skewness < 0:

    print("Interpretation: Negatively skewed (tail on the left)")

else:

    print("Interpretation: Symmetrical distribution")

# Kurtosis

print(f"\nKurtosis: {data\_kurtosis:.4f}")

if data\_kurtosis > 0:

    print("Interpretation: Leptokurtic (sharper peak than normal distribution)")

elif data\_kurtosis < 0:

    print("Interpretation: Platykurtic (flatter than normal distribution)")

else:

    print("Interpretation: Mesokurtic (normal distribution)")

**Output:**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Practical 03**

**Aim: Implement Eigenvalues and EigenVectors.**

**Program:**

import numpy as np

# Function to take matrix input from the user

def input\_matrix(n):

    matrix = []

    print(f"Enter the entries of a {n}x{n} matrix row by row:")

    for i in range(n):

        row = list(map(float, input(f"Row {i+1}: ").split()))

        if len(row) != n:

            print(f"Please enter exactly {n} values.")

            return input\_matrix(n)

        matrix.append(row)

    return np.array(matrix)

# Get the size of the matrix from the user

n = int(input("Enter the size of the square matrix (e.g., 2 for 2x2): "))

# Get the matrix from the user

A = input\_matrix(n)

# Compute eigenvalues and eigenvectors

eigenvalues, eigenvectors = np.linalg.eig(A)

print("\nEigenValues:")

print(eigenvalues)

print("\nEigenVectors:")

print(eigenvectors)

**Output:**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Practical 04**

**Aim: Basics of Statistics:Frequency Table.**

**Program :**from collections import Counter

# Collect from user

entries = []

print("Enter values (type 'done' when finished):")

while True:

    entry = input("Enter an item: ")

    if entry.lower() == 'done':

        break

    entries.append(entry)

# Create a frequency table

frequency\_table = Counter(entries)

# Print the frequency table

print("\nFrequency Table:")

for item, frequency in frequency\_table.items():

    print(f"{item}: {frequency}")

**Output:**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Practical 05**

**Aim: Implement Object-Oriented Programming in Java**

**A) Data Type  
Program:**

public class BasicDatatype {

    public static void main(String[] args) {

        byte byteValue1 = 2;

        byte byteValue2 = 4;

        byte byteResult = (byte) (byteValue1 + byteValue2);

        System.out.println("Byte:" + byteResult);

        short shortValue1 = 2;

        short shortValue2 = 4;

        short shortResult = (short) (shortValue1 + shortValue2);

        System.out.println("Short:" + shortResult);

        int intValue1 = 2;

        int intValue2 = 4;

        int intResult = (int) (intValue1 + intValue2);

        System.out.println("Int:" + intResult);

        long longValue1 = 2L;

        long longValue2 = 4L;

        long longResult = longValue1 + longValue2;

        System.out.println("Long:" + longResult);

        float floatValue1 = 2.0f;

        float floatValue2 = 4.0f;

        float floatResult = floatValue1 + floatValue2;

        System.out.println("Float:" + floatResult);

        double doubleValue1 = 2.0;

        double doubleValue2 = 4.0;

        double doubleResult = doubleValue1 + doubleValue2;

        System.out.println("Double:" + doubleResult);

        boolean booleanValue = true;

        System.out.println("Boolean:" + booleanValue);

        char charValue = 'A';

        System.out.println("Char:" + charValue);

    }

}

**Output:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**B) Simple Calculator  
Program:**

import java.util.Scanner;

public class Calculator {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        // Taking input from the user

        System.out.print("Enter the first and the second number: ");

        int a = sc.nextInt();

        int b = sc.nextInt();

        // Selecting the operand for the calculations

        System.out.print("Choose and Enter the type of operation you want to perform (+, -, \*, /, %): ");

        char op = sc.next().charAt(0);

        solve(a, b, op);

        sc.close();

    }

    private static int solve(int a, int b, char op) {

        int ans = 0;

        // Addition

        if (op == '+') {

            ans = a + b;

        }

        // Subtraction

        else if (op == '-') {

            ans = a - b;

        }

        // Multiplication

        else if (op == '\*') {

            ans = a \* b;

        }

        // Division

        else if (op == '/') {

            ans = a / b;

        }

        // Modulus

        else if (op == '%') {

            ans = a % b;

        }

        // Printing the final result

        System.out.println("Your answer is: " + ans);

        return ans;

    }}

**Output:**

* **Add**

**A black screen with white text

AI-generated content may be incorrect.**

* **Subtract**

**A black background with white text

AI-generated content may be incorrect.**

* **Multiply**

**A black background with white text

AI-generated content may be incorrect.**

* **Divide**

**A black background with white text

AI-generated content may be incorrect.**

* **Mod**

**A black background with white text

AI-generated content may be incorrect.**

**Practical 06**

**Aim: Implement Bubble Sort in Java.**

**Program:**

class BubbleSort {

    public static void main(String[] args) {

        int[] arr = {64, 34, 25, 12, 22, 11, 90};

        int n = arr.length;

        for (int i = 0; i < n - 1; i++) {

            for (int j = 0; j < n - i - 1; j++) {

                if (arr[j] > arr[j + 1]) {

                    // Swap arr[j] and arr[j + 1]

                    int temp = arr[j];

                    arr[j] = arr[j + 1];

                    arr[j + 1] = temp;

                }

            }

        }

        System.out.println("Sorted array:");

        for (int num : arr) {

            System.out.print(num + " ");

        }

    }

}

**Output:**

****

**Practical 07**

**Aim: Implement Selection Sort in Java.**

**Program:**

class SelectionSort {

    public static void main(String[] args) {

        int[] arr = {64, 34, 25, 12, 22, 11, 90};

        int n = arr.length;

        for (int i = 0; i < n - 1; i++) {

            int minIndex = i;

            for (int j = i + 1; j < n; j++) {

                if (arr[j] < arr[minIndex]) {

                    minIndex = j;

                }

            }

            if (minIndex != i) {

                // Swap arr[i] and arr[minIndex]

                int temp = arr[i];

                arr[i] = arr[minIndex];

                arr[minIndex] = temp;

            }

        }

        System.out.println("Sorted array:");

        for (int num : arr) {

            System.out.print(num + " ");

        }

    }

}

**Output:**

**A screenshot of a computer

AI-generated content may be incorrect.**