Practical File
Of
“Artificial Neural Networks Lab”

Subject Code
PC-CS-AIML-315A

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF :

Bachelor of Technology
CSE (Artificial intelligence and Machine Learning)
Session:2022-2026

UNDER THE SUPERVISION OF:
ER. ANJALI THAKUR

SUBMITTED BY:
Keshav
2323982
BRANCH : CSE (AIML)

Department of Computer Science and Engineering

Ambala College of Engineering and Applied Research, Devsthali, Ambala (Haryana)

Affiliated to Kurukshetra University, Kurukshetra

Roll Number - 2323982

Index

Program - 1

AIM - Perform elementary mathematical operations like addition, multiplication, division and exponentiation.

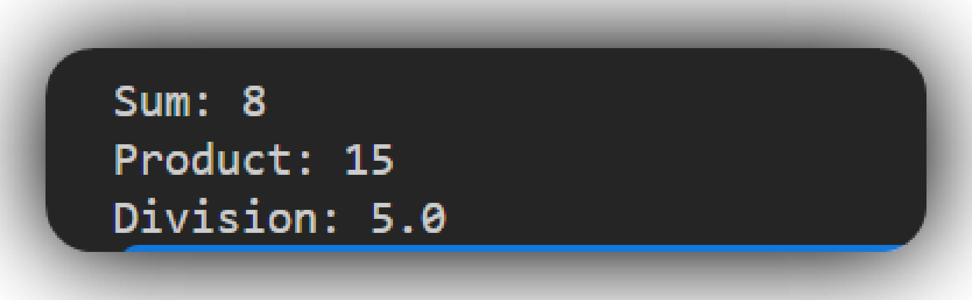
SOFTWARE USED - VS Code

Description: Program initializes two numbers, num1 and num2, adds them together, and stores the result in sum_result. The program then prints the sum. Similar to addition, it initializes two numbers, numl and num2, multiplies them, and stores the product in product_result. The program then prints the product.Division: This code initializes two numbers, numl and num2, divides num1 by num2, and stores the result in division_result. The program then prints the quotient. It initializes a base and an exponent, calculates the result of raising the base to the exponent using the ** operator, and stores it in power_result. The program then prints the result of exponentiation.

SOURCE CODE:

```
# Addition
num1 = 5
num2 = 3
sum_result = num1 + num2
print("Sum:", sum_result)

# Multiplication
numl = 5 num2 = 3 product_result = num1 num2 print("Product:", product_result)

# Division
numl = 10
num2 = 2 division_result = numl / num2
print("Division:", division_result)
```

OUTPUT:



```
Sum: 8
Product: 15
Division: 5.0
```

Program - 2

AIM - Create, initialize, and display simple variables and simple strings and use simple formatting for variable.

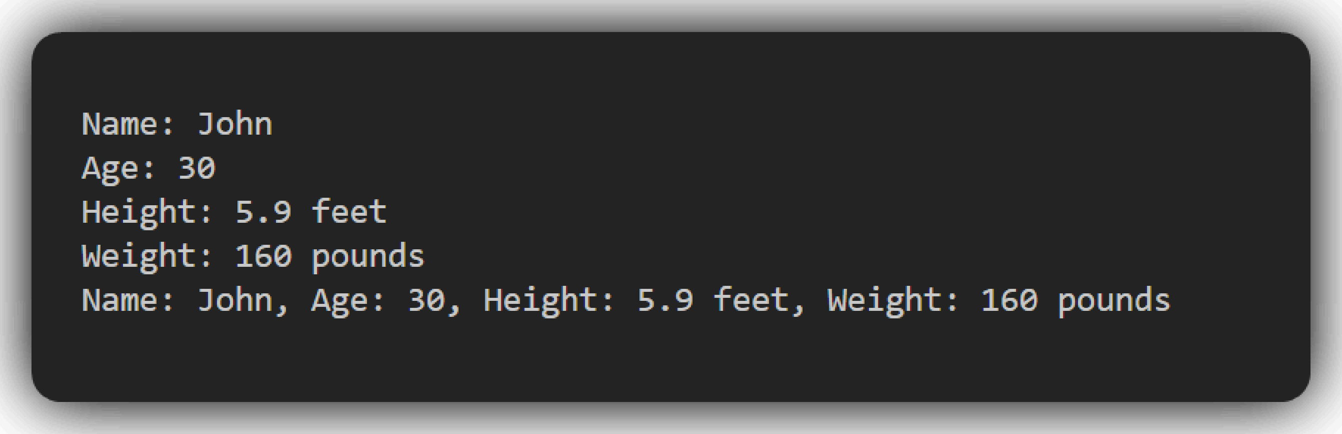
SOFTWARE USED - VS Code

Description: In this program we Create, initialize, and display simple variables and simple strings and use simple formatting for variable. The program demonstrates simple string formatting using the .format() method. It constructs a formatted string, formatted_info, by inserting the values of name, age, height, and weight into placeholders within the string. This formatted string provides a concise way to represent the variable values.

SOURCE CODE:

```
#Creating and Initializing Variables
name = "John"
age = 30
height= 5.9 # in feet
weight= 160 # in pounds
# Displaying Simple Variables
print("Name: " + name)
print("Age: " + str(age))
print("Height: " + str(height) + " feet")
print("Weight: " + str(weight) + " pounds")
# Using Simple Formatting for Variables
formatted_info = "Name: {}, Age: {}, Height: {} feet, Weight: {} pounds".format(name,
age, height, weight)
print(formatted_info)
```

OUTPUT:



```
Name: John
Age: 30
Height: 5.9 feet
Weight: 160 pounds
Name: John, Age: 30, Height: 5.9 feet, Weight: 160 pounds
```

Program - 3

AIM -Create/Define single dimension / multi-dimension arrays, and arrays with specific values like array of all ones, all zeros, array with random values within a range, or a diagonal matrix.

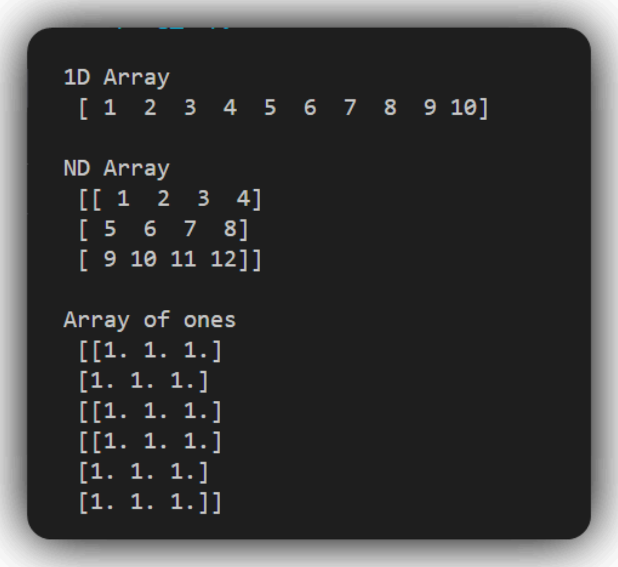
SOFTWARE USED - VS Code

Description: The code uses NumPy functions to generate these arrays. NumPy is a popular library for numerical operations in Python and is especially useful for creating and manipulating arrays. These examples demonstrate different ways to create arrays with specific values and shapes to suit various data processing needs.

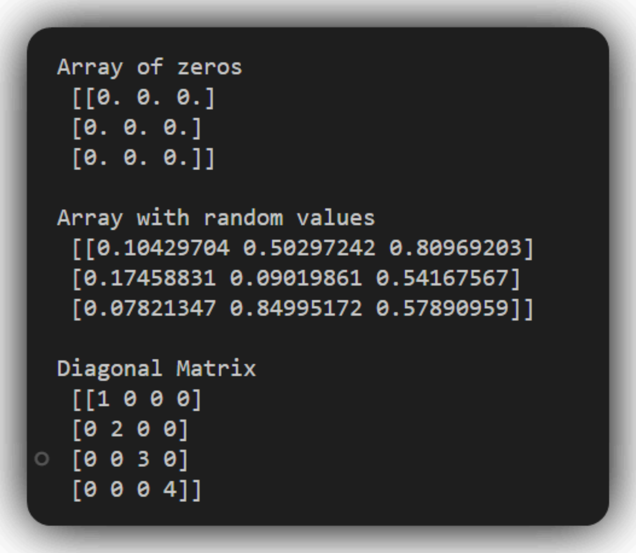
SOURCE CODE:

```
import numpy as np

# Create/Define single dimension / multi-dimension arrays, and arrays with specific
values like array of all ones, all zeros, array with random values within a range, or a
diagonal matrix.

# Single Dimensional
arr_1d = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])

# Multi Dimensional
arr_4d = np.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])

#Array of ones
arr_ones = np.ones((3, 3))

# Arrays of zeros
arr_zeros = np.zeros((3, 3))

# Array with random values
arr_rand = np.random.rand(3, 3)

# Diagonal Matrix
arr_diag = np.diag([1, 2, 3, 4])
print(arr_diag)
```

OUTPUT:



```
1D Array
 [ 1  2  3  4  5  6  7  8  9 10]

ND Array
 [[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]

Array of ones
 [[[1. 1. 1.]
  [1. 1. 1.]]

 [[1. 1. 1.]]

 [[1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]]]
```



```
Array of zeros
 [[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]

Array with random values
 [[0.10429704 0.50297242 0.80969203]
 [0.17458831 0.09019861 0.54167567]
 [0.07821347 0.84995172 0.57890959]]

Diagonal Matrix
 [[1 0 0 0]
 [0 2 0 0]
 [0 0 3 0]
 [0 0 0 4]]
```

Program - 4

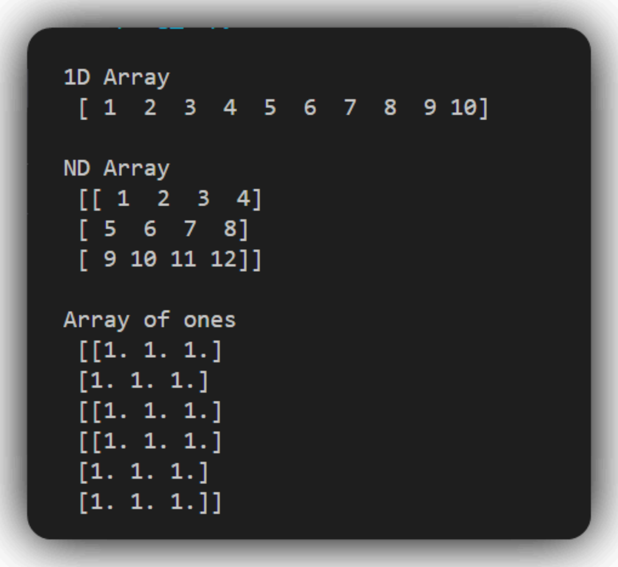
AIM - Use command to compute the size of a matrix, size/length of a particular row/column, load data from a text file, store matrix data to a text file, finding out variables and their features in the current scope

SOFTWARE USED - VS Code

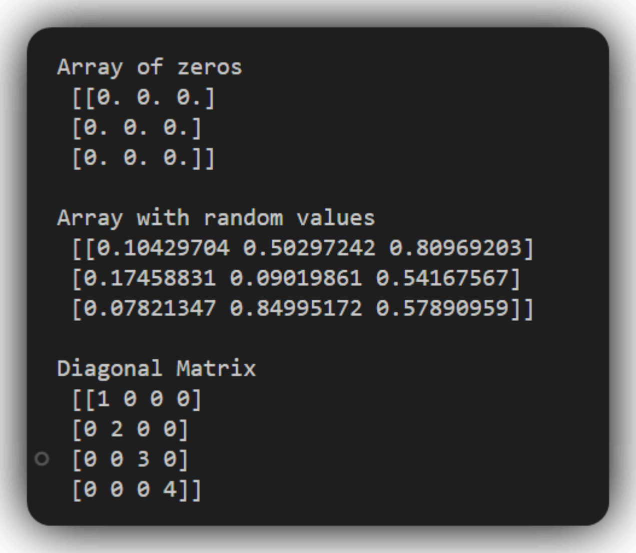
SOURCE CODE:

```
import numpy as np

# Create/Define single dimension / multi-dimension arrays, and arrays with specific
values like array of all ones, all zeros, array with random values within a range, or a
diagonal matrix.
# Single Dimensional
arr_1d = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
# Multi Dimensional
arr_4d = np.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])
#Array of ones
arr_ones = np.ones((3, 3))
# Arrays of zeros
arr_zeros = np.zeros((3, 3))
# Array with random values
arr_rand = np.random.rand(3, 3)
# Diagonal Matrix
arr_diag = np.diag([1, 2, 3, 4])
print(arr_diag)
```

OUTPUT:



```
1D Array
 [ 1  2  3  4  5  6  7  8  9 10]

ND Array
 [[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]

Array of ones
 [[1. 1. 1.]
 [1. 1. 1.]
 [[1. 1. 1.]
 [[1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]]
```



```
Array of zeros
 [[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]

Array with random values
 [[0.10429704 0.50297242 0.80969203]
 [0.17458831 0.09019861 0.54167567]
 [0.07821347 0.84995172 0.57890959]]

Diagonal Matrix
 [[1 0 0 0]
 [0 2 0 0]
 [0 0 3 0]
 [0 0 0 4]]
```