

Python Programming

Lab Questions

4th Semester

Computer Engineering
Department

Prepared By:

Er. Milan Chikanbanjar

Khwopa Engineering College

Purbanchal University

Lab 1

Python Basics

1. Print all the even numbers from 0 to 10.
2. Add logic to print two lines. The first line should contain the result of integer division, //. The second line should contain the result of float division, /.
3. Evaluate following the expression
 - a. $4 * (6 + 5)$
 - b. $4 * 6 + 5$
 - c. $4 + 6 * 5$
 - d. $(5 > 4) \text{ and } (3 == 5)$
 - e. $\text{not } (5 > 4)$
 - f. $(5 > 4) \text{ or } (3 == 5)$
 - g. $\text{not } ((5 > 4) \text{ or } (3 == 5))$
 - h. $(\text{True and True}) \text{ and } (\text{True == False})$
 - i. $(\text{not False}) \text{ or } (\text{not True})$
4. What is the type of the result of the expression $3 * 1.5 + 4$
5. Find a number's square root as well as square
6. Write a Python program to check whether a variable is an integer or string or float.
7. Write a Python program to test if a variable is a list, tuple, or set.
8. Write a Python program to print the following string in a specific format.

Output :

Twinkle, twinkle, little star,
How I wonder what you are!
 Up above the world so high,
 Like a diamond in the sky.

Twinkle, twinkle, little star,
How I wonder what you are

9. Write a Python program to find out what version of Python you are using.
10. Write a Python program to display the first and last colors from the following list.
`color_list = ["Red", "Green", "White", "Black", "Blue"]`
11. Write a Python program to get the volume of a sphere with radius=6.
12. Write a Python program to calculate the sum of three given numbers. If the values are equal, return three times their sum.
13. Write a Python program to sum two given integers. However, if the sum is between 15 and 20 it will return 20.
14. Write a Python program that displays your name, age, and address on three different lines.
15. Write a Python program to solve $(x + y) * (x + y)$.

Test Data : $x = 4, y = 3$

Expected Output : $(4 + 3)^2 = 49$

16. Write a Python program to concatenate N strings.
17. Given variables $x=30$ and $y=20$, write a Python program to print "30+20=50".
18. Print different type of variables i.e. determine the type of an object/ a variable in Python.
19. Write a Python program to ask the user to input their name, stores it in the variable "name", and then prints a greeting message addressing the user by their entered name.
Use `input()` function.
20. Write a Python program to ask the user to input an integer representing the number of roses, converts the input to an integer using typecasting, and then prints the integer value.



Lab 2

Python Basics 2

1. Write a Python program to calculate the difference between a given number and 17. If the number is greater than 17, return twice the absolute difference.
2. Write a Python program that determines whether a given number (accepted from the user) is even or odd, and prints an appropriate message to the user.
3. Write a Python program to count the number 4 in a given list.
4. Write a Python program to swap two variables.
5. Write a Python program to check whether a variable is an integer, or string, or a list, or tuple, or set.
6. Write a Python script that takes two numbers as input and prints their sum, difference, product, and quotient.
7. Take an input from user then reverse the string using slicing.
8. Write code to take input from user and store it in a variable spam then print Hello if 1 is stored in spam, print Hi if 2 is stored in spam, and print Greetings! if anything else is stored in spam.
9. Write a Python script that takes two numbers as input and prints their sum, difference, product, and quotient using match case.
10. Write a script that asks the user for their name and age, then prints a message that tells them the year in which they will turn 100 years old.
11. Create a Python script that converts temperature from Fahrenheit to Celsius and vice versa, based on user input
12. Create a program that asks for an age and prints "Child" if the age is less than 12, "Teenager" if the age is between 13 and 19, and "Adult" for ages 20 and above
13. Write a python script that takes a letter grade (A, B, C, D, F) as input and prints the corresponding grade point average (GPA). For example, A = 4.0, B = 3.0, C = 2.0, D = 1.0, F=0.0. Include an else statement to handle invalid inputs.
14. Write a python program that takes a number and prints whether it is "Even", "Odd", "Zero", or "Invalid" for non-integer inputs. This program should first check if the input is a valid integer and then check for the other conditions.
15. An extra day is added to the calendar almost every four years as February 29, and the day is called a leap day. It corrects the calendar for the fact that our planet takes approximately 365.25 days to orbit the sun. A leap year contains a leap day.

In the Gregorian calendar, three conditions are used to identify leap years:

- The year can be evenly divided by 4, is a leap year, unless:
- The year can be evenly divided by 100, it is not a leap year, unless:
- The year is also evenly divisible by 400. Then it is a leap year.

This means that in the Gregorian calendar, the years 2000 and 2400 are leap years, while 1800, 1900, 2100, 2200, 2300 and 2500 are not leap years.

Given a year, determine whether it is a leap year. If it is a leap year, return the Boolean True otherwise return False.



Lab 3

Python Control Statement and Loops

1. Write a python program to find the sum of n natural numbers
2. Write a program to read an integer from user. For all non-negative integers $i < n$, print i^2 .
3. Write a Python program that prints all the numbers from 0 to 6 except 3 and 6.

Note: Use 'continue' statement.

Expected Output : 0 1 2 4 5

4. Write a Python program to get the Fibonacci series up to n terms.
5. Write a Python program to find those numbers which are divisible by 7 and multiples of 5, between 1500 and 2700 (both included).

6. Write a Python program to guess a number between 1 and 9.

Note : User is prompted to enter a guess. If the user guesses wrong then the prompt appears again until the guess is correct, on successful guess, user will get a "Well guessed!" message, and the program will exit. *Import random.*

7. Write a Python program to input n numbers from user; count the number of even and odd numbers from the input numbers and exit until the user input '0'.

Input numbers = (7, 4, 3, 8, 5, 6, 1, 2, 9)

Expected Output :

Number of even numbers : 4

Number of odd numbers : 5

8. Write a Python program that accepts a string and calculates the number of digits and letters.

Sample Data : Python 3.2

Expected Output :

Letters 6

Digits 2

Note: Use *isdigit*, *isalpha* and *pass* functions.

9. Write a Python program to develop a rock paper scissors game, restart the game until the user press 'n' when the game ends.

10. Write a Python program to create the multiplication table (from 1 to 10) of a number.

Expected Output:

Input a number: 6

$6 \times 1 = 6$

$6 \times 2 = 12$

$6 \times 3 = 18$

$6 \times 4 = 24$

$6 \times 5 = 30$

$6 \times 6 = 36$

$6 \times 7 = 42$

$6 \times 8 = 48$

$6 \times 9 = 54$

$6 \times 10 = 60$

11. Write a Python program that accepts a word from the user and reverses it.

Jalpa

12. Write a Python program to construct the following pattern, using a nested for loop.

```
*  
* *  
* * *  
* * * *  
* * * * *  
* * * *  
* * *  
* *  
*
```



Lab 4

Python Lists

1. Write a Python Program to print all the items in a list.
2. Use range function to print all the even numbers from 1 to 10.
3. Write a Python Program to get the largest and smallest number in a list without using built-in functions.
4. Write a Python program to find the second smallest and second largest numbers in a list.
5. Write a Python program to sum all the items in a list.
6. Write a Python program to multiply all the items in a list.
7. Write a Python program to check if a list is empty or not.
8. Write a Python program to print the numbers of a specified list after removing even numbers from it.
9. Write a Python Program to print a specified list after removing the 0th, 4th and 5th element.

Sample list: ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']

Expected Output: ['Green', 'White', 'Black']

10. Write a Python Program to check if each number is prime in a given list of numbers. Print the prime numbers if any, otherwise print "no prime numbers".
11. Write a Python program to remove duplicates from a list.
12. Write a Python Program to merge two lists and removes all duplicates from the combined list.
13. Write a Python Program to print a list in a reverse order.
14. Write a Python program to sort a given list of strings (numbers) numerically.

Original list:

[4, '12', '45', '7', '0', '100', '200', '-12', '-500']

Sort the said list of strings(numbers) numerically:

[-500, -12, 0, 4, 7, 12, 45, 100, 200]

15. Apply various functions like append, remove, pop, insert, sort, max, min in a list.
16. Write a python program to copy a content of one list to another.
17. Write a Python Program to sort a list of lists by a given index of the inner list.
18. Write a Python program to generate and print a list of the first and last 5 elements where the values are square numbers between 1 and 15 (both included).
19. Write a Python program to insert a given string at the beginning of all items in a list.

Sample list : [1,2,3,4], string : emp

Expected output : ['emp1', 'emp2', 'emp3', 'emp4']

20. Write a Python program to insert n values in a list and find those values in a list that are greater than a specified number.

Fallo

Lab 5

Python Tuples and Sets

Tuples

1. Write a Python program to convert a tuple to a string.
2. Write a Python program to get the 4th element from the last element of a tuple.
3. Write a Python program to find repeated items in a tuple.
4. Write a Python program to replace the last value of tuples in a list.

Sample list: [(10, 20, 40), (40, 50, 60), (70, 80, 90)]

Expected Output: [(10, 20, 100), (40, 50, 100), (70, 80, 100)]

5. Write a Python program to sort a tuple by its float element.

Sample data: [('item1', '12.20'), ('item2', '15.10'), ('item3', '24.5')]

Expected Output: [('item3', '24.5'), ('item2', '15.10'), ('item1', '12.20')]

6. Write a Python program to convert a given string list to a tuple.

Original string: python 3.0

<class 'str'>

Convert the said string to a tuple:

('p', 'y', 't', 'h', 'o', 'n', '3', '!', '0')

<class 'tuple'>

7. Write a Python program to calculate the product, multiplying all the numbers in a given tuple.

Original Tuple:

(4, 3, 2, 2, -1, 18)

Product - multiplying all the numbers of the said tuple: -864

Original Tuple:

(2, 4, 8, 8, 3, 2, 9)

Product - multiplying all the numbers of the said tuple: 27648

8. Write a Python program to calculate the average value of the numbers in a given tuple of tuples.

Original Tuple:

((10, 10, 10, 12), (30, 45, 56, 45), (81, 80, 39, 32), (1, 2, 3, 4))

Average value of the numbers of the said tuple of tuples:

[30.5, 34.25, 27.0, 23.25]

Original Tuple:

((1, 1, -5), (30, -15, 56), (81, -60, -39), (-10, 2, 3))

Average value of the numbers of the said tuple of tuples:

[25.5, -18.0, 3.75]

9. Write a Python program to convert a given tuple of positive integers into an integer.

Original tuple:

(1, 2, 3)

Convert the said tuple of positive integers into an integer:

[Signature]

12³

Original tuple:

(10, 20, 40, 5, 70)

Convert the said tuple of positive integers into an integer:

102040570

10. Write a Python program to compute the element-wise sum of given tuples.

Original lists:

(1, 2, 3, 4)

(3, 5, 2, 1)

(2, 2, 3, 1)

Element-wise sum of the said tuples:

(6, 9, 8, 6)

11. Write a Python program to compute the sum of all the elements of each tuple stored inside a list of tuples.

Original list of tuples:

[(1, 2), (2, 3), (3, 4)]

Sum of all the elements of each tuple stored inside the said list of tuples:

[3, 5, 7]

Original list of tuples:

[(1, 2, 6), (2, 3, -6), (3, 4), (2, 2, 2, 2)]

Sum of all the elements of each tuple stored inside the said list of tuples:

[9, -1, 7, 8]

Sets

12. Write a Python program to create an intersection, a union, set difference and a symmetric difference of sets. Also find the length, maximum and minimum values in a set.

13. Write a Python program that finds all pairs of elements in a list whose sum is equal to a given value.

14. Write a Python program to find the longest common prefix of all strings. Use the Python set.

15. Write a Python program to find the two numbers whose product is maximum among all the pairs in a given list of numbers. Use the Python set.

Felicia

Lab 6

Python Dictionary

1. Write a Python program to create a python dictionary with the following key-value pairs: 'name': 'Hari', 'age': '29', 'city': 'Bhaktapur'.
2. Write a Python program to add a key-value pair 'occupation': 'Engineer' into an existing dictionary.
3. In the above dictionary, change the value of 'city' to 'Kathmandu'.
4. Write a Python program to remove a key 'city' from the dictionary.
5. Write a python program to loop to iterate through the dictionary from the first question and print all keys and their corresponding values.
6. Write a Python program to get the key, value and item in a dictionary.
7. Write a python script to check whether the key 'email' exists in the dictionary. If it does not exist print 'key not found'.
8. Write a Python script to merge two Python dictionaries.
dict1 = {'a': 1, 'b': 2} and dict2 = {'b': 3, 'c': 4} merge them into a single dictionary. What happens to the value of the key 'b'?
9. Create a script that converts a list of tuples, [('a', 1), ('b', 2), ('c', 3)], into a dictionary.
10. Use the get() method to fetch the value of 'age' from the above dictionary. If the 'age' is not a key, it should return 'Key not available'.
11. Write a Python program to find all keys in a dictionary that have the given value.
12. Write a Python program to sort a dictionary by its keys.
13. Write a Python program to sort a dictionary by values instead of keys.
14. Write a Python script to check whether a given key already exists in a dictionary.
15. Write a Python program to sort a list alphabetically in a dictionary.
Sample: num_dict = {'n1': [2, 3, 1], 'n2': [5, 1, 2], 'n3': [3, 2, 4]}
Output: {'n1': [1, 2, 3], 'n2': [1, 2, 5], 'n3': [2, 3, 4]}
16. Given a dictionary d = {'a': 1, 'b': 2, 'c': 3}, write a python script to reverse the key-value pairs to get a new dictionary.
17. Write a Python program to create a dictionary from a string.
Note: Track the count of the letters from the string.
Sample string : 'engineering'
Expected output: {'e': 3, 'n': 2, 'g': 2, 'i': 2, 'r': 1, 's': 1}
18. Write a Python program to find the highest 3 values of corresponding keys in a dictionary.
19. Write a Python program to drop empty items from a given dictionary.
Original Dictionary: {'c1': 'Red', 'c2': 'Green', 'c3': None}
New Dictionary after dropping empty items: {'c1': 'Red', 'c2': 'Green'}
20. Write a Python program to find the specified number of maximum values in a given dictionary.
Original Dictionary: {'a': 5, 'b': 14, 'c': 32, 'd': 35, 'e': 24, 'f': 100, 'g': 57, 'h': 8, 'i': 100}
1 maximum value(s) in the said dictionary: ['f']
2 maximum value(s) in the said dictionary: ['f', 'i']
5 maximum value(s) in the said dictionary: ['f', 'i', 'g', 'd', 'c']
21. Write a Python program to find the key of the maximum and minimum values in a dictionary.
Original Dictionary: {'a': 5, 'b': 14, 'c': 32, 'd': 35}
Maximum and minimum key values: ('d', 'a')

Milan

Lab 7

Python function

1. Create a function named "greet" that prints "Hello World!".
2. Write a Python function to find the maximum of three input numbers.
3. Write a Python function to calculate the factorial of a number (a non-negative integer) with and without using recursion. The function accepts the number as an argument.
4. Write a python function that takes two parameters length and width and returns the area and perimeter of a rectangle.
5. Modify above function so that it has default values of 2 for both length and width.
6. Write a python function that accepts radius and returns the area and circumference of a circle. Import Pi from Math.
7. Write a function that takes two arguments, name and age and returns a dictionary with these as keys and their respective values.
8. Write a Python function to find the maximum and minimum value, sum and multiplication of all the numbers in a list.
9. Write a python program to demonstrate positional argument and functional arguments in function.
10. Create a function that takes a list and a number, return a list after adding the number to the list preventing it from changing the original list.
11. Write a function that takes a list of numbers and print each number doubled.
12. Defines a function called calculate_average that takes a list of numbers as input and calculate the average of list. Finally, the function returns the average of that list.
13. Write a Python function that checks whether a passed string is a palindrome or not.
14. Write a Python function that takes a string as input and counts the number of uppercase and lowercase characters in the string.
15. Write a Python recursive function to find out factorial of any given number.



Lab 8

Python functions 2

1. Write a Python program to reverse a string.
2. Write a Python function to check whether a number falls within a given range.
3. Write a Python function that accepts a string and counts the number of vowel and consonant letters.
4. Write a Python function that takes a list and returns a new list with distinct elements from the first list.
5. Write a Python function to create and print a list where the values are the squares of numbers between 1 and 20 (both included).
6. Write a Python program to access a function inside a function.
7. Write a Python program to detect the number of local variables declared in a function.
8. Write a Python program to count the even and odd numbers from a given list and also print them separately.
9. Write a Python program to find the power of a number using recursion function.

Input: N = 2, P = 3 **Output:** 8

Input: N = 5, P = 2 **Output:** 25

10. Write a Python function that takes a sentence as a parameter and print the words in ascending order.
11. Write a Python a function that takes a string as argument and print the most common character in that string.
12. Write a Python function that takes a date in string format DD/MM/YYYY and checks if it is a valid date and in the correct format.
13. Write a Python function that takes a name (string) as argument and capitalizes the first and fourth letters of the input name.
14. Write a Python function that takes a sentence, and return a sentence with the words reversed.
15. Write a Python function takes a two-word strings and find if both words begin with same letter or not.

[Signature]

Lab 9

Python Class

1. Define a Python function `student()`. Using function attributes display the names of all arguments.
2. Write a Python function `student_data()` that will print the ID of a student (`student_id`). If the user passes an argument `student_name` or `student_class` the function will print the student name and class.
3. Write a Python class named `Student` with two attributes `student_name`, `marks`. Modify the attribute values of the said class and print the original and modified values of the said attributes.
4. Write a Python class named `Student` with two attributes `student_id`, `student_name`. Add a new attribute `student_class` and display the entire attribute and the values of the class. Now remove the `student_name` attribute and display the entire attribute with values.
5. Write a Python class named `Student` with two attributes: `student_id`, `student_name`. Add a new attribute: `student_class`. Create a function to display all attributes and their values in the `Student` class.
6. Write a Python class named `Dog` that has two attributes: `name` and `age`. Then, create an instance of your `Dog` class and print out the name and age of the dog.
7. Create two instances from the `Dog` class. For the first instance, set the name to 'Bruno' and age to 2. For the second instance, set the name to 'Sher' and age to 3. Print out the information for both dogs.
8. Define a class named `Car` with attributes `make`, `model`, and `year`. Add a method `describe_car()` that prints a neatly formatted descriptive name of the car. Then, create an instance of `Car` and call this method.
9. Modify the `Car` class so that it has a default value for an attribute called `fuel_level`, with a default value of 100. Add a method `check_fuel_level()` that prints the car's current fuel level.
10. Add a method to the `Car` class that allows you to update the car's `fuel_level`. Then, create an instance of `Car`, update its `fuel_level` to 50, and call `check_fuel_level()` to verify the change.
11. Create a child class called `ElectricCar` that inherits from the `Car` class. Add an attribute `battery_size` to the child class with a default value. Also, add a method `describe_battery()` that prints information about the battery size.
12. Modify the `ElectricCar` class to include an `__init__()` method that properly initializes the parent class's attributes as well as its own attribute `battery_size`.
13. Add a new method to the `ElectricCar` class that calculates the range of the car based on the battery size. The method should print a message indicating the range.
14. Override the `describe_car()` method in the `ElectricCar` class to include information about its battery size along with the car's make, model, and year.
15. Assume you have another file called `car.py` that contains the `Car` class. Write a new Python script that imports the `Car` class from `car.py`, creates an instance of `Car`, and calls the `describe_car()` method.

Falak

Lab 10

Python Files

1. Write a function in python to read the content from a text file "poem.txt" line by line and display the same on screen
2. Write a Python program to read first n lines of a file.
3. Write a Python program to count the number of lines in a text file which starts with an alphabet "T".
4. Write a Python program to append text to a file and display the text.
5. Write a Python program to read a file line by line and store it into a list.
6. Write a Python program to read a file line by line store it into a variable.
7. Write a Python program to read a file line by line store it into an array.
8. Write a Python program to count and display the total number of words in a file.
9. Write a function in Python to read lines from a text file "notes.txt". Your function should find and display the occurrence of the word "the".
10. Write a Python program to write a list to a file.
11. Write a Python program to copy the contents of a file to another file.
12. Write a Python program to read a random line from a file.
13. Write a Python program to remove newline characters from a file.
14. Write a Python program to extract characters from various text files and puts them into a list.
15. Ram has used a text editing software to type some text. After saving the article as WORDS.TXT, she realized that she has wrongly typed alphabet J in place of alphabet I everywhere in the article. Write a function definition for JTOI() in Python that would display the corrected version of entire content of the file WORDS.TXT with all the alphabets "J" to be displayed as an alphabet "I" on screen.
Note: Assuming that WORD.TXT does not contain any J alphabet otherwise.
16. Demonstrate Renaming, Moving, Copying, and Removing operations of Files in python with or without shutil package.

Kelkar

Lab 11 Machine Learning

1. Load a CSV file into a Pandas DataFrame.

```
import pandas as pd
file_path = 'test.csv'
df = pd.read_csv(file_path)
print(df)
```

2. Create a DataFrame from a NumPy array with custom column names.

```
import pandas as pd
import numpy as np
numpy_array = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
column_names = ['Column1', 'Column2', 'Column3']
df = pd.DataFrame(data=numpy_array, columns=column_names)
print(df)
```

3. Select rows from a DataFrame based on multiple conditions.

```
import pandas as pd
data = {'Name': ['Serij', 'Ram', 'Rabin', 'Santosh', 'Milan'],
        'Age': [26, 32, 25, 31, 28],
        'Salary': [50000, 60000, 45000, 70000, 55000]}
df = pd.DataFrame(data)
selected_rows = df[(df['Age'] > 25) & (df['Salary'] > 50000)]
print(selected_rows)
```

4. Select the first and last 7 rows of a Pandas DataFrame.

```
import pandas as pd
data = {'Name': ['Serij', 'Ram', 'Rabin', 'Santosh', 'Milan', 'Alina', 'Bijay', 'Chandra', 'Dinesh',
               'Suman'],
        'Age': [26, 32, 25, 31, 28, 22, 35, 30, 40, 28],
        'Salary': [50000, 60000, 45000, 70000, 55000, 60000, 70000, 55000, 75000, 65000]}
df = pd.DataFrame(data)
selected_rows = pd.concat([df.head(7)])
print("First 7 rows:")
print(selected_rows)
selected_rows = pd.concat([df.tail(7)])
print("\nLast 7 rows:")
print(selected_rows)
```

5. Merge two Pandas DataFrames based on a common column.

```
import pandas as pd
df1 = pd.DataFrame({'ID': [1, 2, 3], 'Name': ['Rabin', 'Santosh', 'Milan']})
df2 = pd.DataFrame({'ID': [2, 3, 4], 'Age': [25, 30, 22]})
merged_df = pd.merge(df1, df2, on='ID')
print(merged_df)
```

[Signature]

6. Extract rows from a Pandas DataFrame where a specific column's values are in a given NumPy array.

```
import pandas as pd
import numpy as np
data = {'Name': ['Teodosija', 'Sutton', 'Taneli', 'David', 'Emily'],
        'Age': [25, 30, 22, 35, 28],
        'Salary': [50000, 60000, 45000, 70000, 55000]}
df = pd.DataFrame(data)
selected_age_values = np.array([25, 35])
selected_rows = df[df['Age'].isin(selected_age_values)]
print(selected_rows)
```

7. Perform element-wise addition of a NumPy array and a Pandas DataFrame column.

```
import pandas as pd
import numpy as np
data = {'Name': ['Serij', 'Ram', 'Rabin', 'Santosh', 'Milan'],
        'Age': [25, 30, 22, 35, 28],
        'Salary': [50000, 60000, 45000, 70000, 55000]}
df = pd.DataFrame(data)
numpy_array = np.array([1000, 2000, 3000, 4000, 5000])
df['Updated_Salary'] = np.add(df['Salary'], numpy_array)
print(df)
```

8. Calculate the cumulative sum of a NumPy array and store the results in a new Pandas DataFrame column.

```
import pandas as pd
import numpy as np
data = {'Values': [100, 200, 300, 400, 500]}
df = pd.DataFrame(data)
numpy_array = np.array(df['Values'])
df['Cumulative_Sum'] = np.cumsum(numpy_array)
print(df)
```

9. Create a histogram of a numerical column using NumPy and Matplotlib.

```
import numpy as np
import matplotlib.pyplot as plt
data = np.random.randn(1000) # Generating random data for demonstration
hist, edges = np.histogram(data, bins=10)
plt.hist(data, bins=edges, edgecolor='black', alpha=0.7)
plt.title('Histogram of a Numerical Column')
plt.xlabel('Values')
plt.ylabel('Frequency')
plt.show()
```

10. Perform matrix multiplication using NumPy.

```
import numpy as np
matrix1 = np.array([[1, 2, 3], [4, 5, 6]])
```

S. J. Chikkanjar

```
matrix2 = np.array([[7, 8], [9, -10], [11, 12]])
result_matrix = np.dot(matrix1, matrix2)
print(result_matrix)
```

11. Write a Python program that creates a TensorFlow constant with the values [100, 200, 300].
Print its shape and data type.

```
import tensorflow as tf
t = tf.constant([100, 200, 300])
print("Tensor Shape:", t.shape)
print("Data Type:", t.dtype)
```

12. Write a Python program that creates a TensorFlow tensor with a string data type and print its value.

```
import tensorflow as tf
string_tensor = tf.constant("TensorFlow Exercises!", dtype=tf.string)
print("String Tensor Value:", string_tensor.numpy().decode())
```