

PYTHON PROGRAMMING LAB



Prepared by:

Name of Student: _Shikha singh_____

Roll No: 25_____

Batch: 2023-27

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Ex p. No	List of Experiment
1	1.1 Write a program to compute Simple Interest.

	1.2 Write a program to perform arithmetic, Relational operators.
	1.3 Write a program to find whether a given no is even & odd.
	1.4 Write a program to print first n natural number & their sum.
	1.5 Write a program to determine whether the character entered is a Vowel or not .
	1.6 Write a program to find whether given number is an Armstrong Number.
	1.7 Write a program using for loop to calculate factorial of a No.
	1.8 Write a program to print the following pattern
	i) * * * * * * * * * * * * * * *
	ii) 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5
	iii) * * * * * * * * * * * * * * * * * * * * * * * *

2	2.1 Write a program that define the list of defines the list of define countries that are in BRICS.
	2.2 Write a program to traverse a list in reverse order. 1.By using Reverse method. 2.By using slicing
	2.3 Write a program that scans the email address and forms a tuple of username and domain.
	2.4 Write a program to create a list of tuples from given list having number and add its cube in tuple. i/p: c= [2,3,4,5,6,7,8,9]
	2.5 Write a program to compare two dictionaries in Python? (By using == operator)
	2.6 Write a program that creates dictionary of cube of odd numbers in the range.
	2.7 Write a program for various list slicing operation. a= [10,20,30,40,50,60,70,80,90,100] i. Print Complete list ii. Print 4th element of list iii. Print list from 0th to 4th index. iv. Print list -7th to 3rd element v. Appending an element to list. vi. Sorting the element of list. vii. Popping an element. viii. Removing Specified element. ix. Entering an element at specified index. x. Counting the occurrence of a specified element. xi. Extending list. xii. Reversing the list.
3	3.1 Write a program to extend a list in python by using given approach. i. By using + operator. ii. By using Append () iii. By using extend ()

	3.2 Write a program to add two matrices.
	3.3 Write a Python function that takes a list and returns a new list with distinct elements from the first list.
	3.4 Write a program to Check whether a number is perfect or not.
	3.5 Write a Python function that accepts a string and counts the number of upper and lower-case letters. string_test= 'Today is My Best Day'
4	4.1 Write a program to Create Employee Class & add methods to get employee details & print.
	4.2 Write a program to take input as name, email & age from user using combination of keywords argument and positional arguments (*args and **kwargs) using function,
	4.3 Write a program to admit the students in the different Departments(pgdm/btech)and count the students. (Class, Object and Constructor).
	4.4 Write a program that has a class store which keeps the record of code and price of product display the menu of all product and prompt to enter the quantity of each item required and finally generate the bill and display the total amount.
	4.5 Write a program to take input from user for addition of two numbers using (single inheritance).
	4.6 Write a program to create two base classes LU and ITM and one derived class. (Multiple inheritance).
	4.7 Write a program to implement Multilevel inheritance, Grandfather → Father- → Child to show property inheritance from grandfather to child.

	4.8 Write a program Design the Library catalogue system using inheritance take base class (library item) and derived class (Book, DVD & Journal) Each derived class should have unique attribute and methods and system should support Check in and check out the system. (Using Inheritance and Method overriding)
--	---

5	5.1 Write a program to create my_module for addition of two numbers and import it in main script.
	5.2 Write a program to create the Bank Module to perform the operations such as Check the Balance, withdraw and deposit the money in bank account and import the module in main file.
	5.3 Write a program to create a package with name cars and add different modules (such as BMW, AUDI, NISSAN) having classes and functionality and import them in main file cars.
6	6.1 Write a program to implement Multithreading. Printing “Hello” with one thread & printing “Hi” with another thread.
7.	7.1 Write a program to use ‘whether API’ and print temperature of any city, also print the sunrise and sunset times for the same humidity of that area.
	7.2 Write a program to use the ‘API’ of crypto currency.

Name of Student: Shikha singh

_____ **Roll Number: 25**

_____ **Experiment No:**

7.1

Title:

7.1 Write a program to use ‘whether API’ and print temperature of any city, also

print the sunrise and sunset times for the same humidity of that area.

Theory: API Key:

- An API key is a unique identifier that allows the code to authenticate and access the OpenWeatherMap API. This key is necessary for making requests to the weather service.

2. API Request:

- The `requests.get()` function is used to send an HTTP GET request to the OpenWeatherMap API. The URL includes the city name and the API key for authentication.

3. JSON Response:

- The response from the API is in JSON format, which is a lightweight data interchange format. The `response.json()` method is used to convert the JSON data into a Python dictionary.

4. User Input:

- The `input()` function is used to get user input for the city name. This allows the user to specify which city's weather information they want to retrieve.

5. Error Handling:

- The code checks if the API response contains a 'message' key. If present, it indicates that the city was not found, and an error message is displayed.

6. Data Display:

- If the city is found, the code extracts and prints various weather-related information from the JSON response, such as temperature, humidity, sunrise time, and sunset time.


7. Timestamp Conversion:

- The `datetime` module is used to convert UNIX timestamps (received from the API) to human-readable date and time.

Code: api_key="6126a5d84240bd19cddc584c3a690750"

```
import requests
import datetime
city=input("Enter city: ")
response=requests.get(f"https://api.openweathermap.org/data/2.5/weather?q={city}&APPID={api_key}&units=Metric"
)
a=response.json()
if 'message' in a:
    print("City not Found!")
else:
    print("\nCity:",city)
    print("Temperature:",a['main']['temp'], "C")
    print("Humidity:",a['main']['humidity'])
    print("Sunrise(IST):",datetime.datetime.fromtimestamp(a['sys']['sunrise']))
    print("Sunset(IST):",datetime.datetime.fromtimestamp(a['sys']['sunset']))
```

Output: (screenshot)



```
Enter city: mumbai

City: mumbai
Temperature: 26.99 C
Humidity: 69
Sunrise(IST): 2024-01-01 07:11:31
Sunset(IST): 2024-01-01 18:11:43
```

Test Case: Any two (screenshot)

```
Enter city: mumbai  
City: mumbai  
Temperature: 26.99 C  
Humidity: 69  
Sunrise(IST): 2024-01-01 07:11:31  
Sunset(IST): 2024-01-01 18:11:43
```

```
Enter city: surat  
City: surat  
Temperature: 22.99 C  
Humidity: 60  
Sunrise(IST): 2024-01-01 07:15:43  
Sunset(IST): 2024-01-01 18:07:38  
○ shikhasingh@SHIKHAS-MacBook-Air requests %
```

Conclusion: The code showcases the

integration of an external weather API, user

interaction, handling JSON responses, error

checking, and presenting weather

information in a readable format. It

provides a practical example of making API

requests and processing the received data in

a Python script.

