PART- A

Q1) Aim: Introduce the Python fundamentals, data types, operators, flow control and exception handling in Python

a) Write a python program to find the best of two test average marks out of three test’s marks accepted from the user.

m1 = int(input("Enter marks for test1 : "))  
m2 = int(input("Enter marks for test2 : "))  
m3 = int(input("Enter marks for test3 : "))  
  
if m1 <= m2 and m1 <= m3:  
 avgMarks = (m2+m3)/2  
elif m2 <= m1 and m2 <= m3:  
 avgMarks = (m1+m3)/2  
elif m3 <= m1 and m2 <= m2:  
 avgMarks = (m1+m2)/2   
   
print("Average of best two test marks out of three test’s marks is", avgMarks);

**Output:**

Enter marks for test1 : 92

Enter marks for test2 : 89

Enter marks for test3 : 92

Average of best two test marks out of three test’s marks is 92.0

b) Develop a Python program to check whether a given number is palindrome or not and also count the number of occurrences of each digit in the input number.

print("method 1")  
str\_val = input("Enter a value : ")  
if str\_val == str\_val[::-1]:  
 print("Palindrome")  
else:  
 print("Not Palindrome")  
  
for i in range(10):  
 if str\_val.count(str(i)) > 0:  
 print(str(i),"appears", str\_val.count(str(i)), "times");

**Output:**Enter a value : 5657565

Palindrome

5 appears 4 times

6 appears 2 times

7 appears 1 times  
  
Enter a value : 5686

Not Palindrome

5 appears 1 times

6 appears 2 times

8 appears 1 times

Q2) Aim: Demonstrating creation of functions, passing parameters and return values

Fibonacci Sequence

1. Defined as a function F as Fn = Fn-1 + Fn-2. Write a Python program which accepts a value for N (where N >0) as input and pass this value to the function. Display suitable error message if the condition for input value is not followed.

def Fn(n):  
 if n <= 1:  
 return n  
 else:  
 return(Fn (n-1) + Fn (n-2))  
  
nterms = int(input("enter the number"))  
  
*# check if the number of terms is valid*if nterms <= 0:  
 print("Plese enter a positive integer")  
else:  
 print("Fibonacci sequence:")  
 for i in range(nterms):  
 print(Fn (i))

**Output:**  
enter the number10

Fibonacci sequence:

1

1

2

3

5

8

13

21

34

1. Develop a python program to convert binary to decimal, octal to hexadecimal using functions.

def bin2Dec(val):  
 rev=val[::-1]  
 dec = 0  
 i = 0  
 for dig in rev:  
 dec += int(dig) \* 2\*\*i  
 i += 1  
 return dec  
  
def oct2Hex(val):  
 rev=val[::-1]  
 dec = 0  
 i = 0  
 for dig in rev:  
 dec += int(dig) \* 8\*\*i  
 i += 1  
 list=[]  
 while dec != 0:  
 list.append(dec%16)  
 dec = dec // 16  
   
 nl=[]  
 for elem in list[::-1]:  
 if elem <= 9:  
 nl.append(str(elem))  
 else:  
 nl.append(chr(ord('A') + (elem -10)))  
 hex = "".join(nl)  
 return hex  
  
num1 = input("Enter a binary number : ")   
print(bin2Dec(num1))  
num2 = input("Enter a octal number : ")  
print(oct2Hex(num2))

**Output:**

Enter a binary number : 10111001

185

Enter a octal number : 675

1BD

Sentence Statistics

Q3) Aim: Demonstration of manipulation of strings using string methods

1. Write a Python program that accepts a sentence and find the number of words, digits, uppercase letters and lowercase letters.

def bin2Dec(val):  
 rev=val[::-1]  
 dec = 0  
 i = 0  
 for dig in rev:  
 dec += int(dig) \* 2\*\*i  
 i += 1  
 return dec  
  
def oct2Hex(val):  
 rev=val[::-1]  
 dec = 0  
 i = 0  
 for dig in rev:  
 dec += int(dig) \* 8\*\*i  
 i += 1  
 list=[]  
 while dec != 0:  
 list.append(dec%16)  
 dec = dec // 16  
   
 nl=[]  
 for elem in list[::-1]:  
 if elem <= 9:  
 nl.append(str(elem))  
 else:  
 nl.append(chr(ord('A') + (elem -10)))  
 hex = "".join(nl)  
 return hex  
  
num1 = input("Enter a binary number : ")   
print(bin2Dec(num1))  
num2 = input("Enter a octal number : ")  
print(oct2Hex(num2))

**Output:**

Enter a sentence : Rama went to Devaraja market to pick 2 kgs of vegetable

This sentence has 11 words

This sentence has 1 digits 2 upper case letters 42 lower case letters

b) Write a Python program to find the string similarity between two given strings

Sample Output: Sample Output: Original string: Original string:

Python Exercises Python Exercises

Python Exercises Python Exercise

Similarity between two said strings: Similarity between two said strings:

1.0 0.967741935483871

str1 = input(**"Enter String 1 \n"**)  
str2 = input(**"Enter String 2 \n"**)  
  
**if** len(str2) < len(str1):  
 short = len(str2)  
 long = len(str1)  
**else**:  
 short = len(str1)  
 long = len(str2)  
  
matchCnt = 0  
**for** i **in** range(short):  
 **if** str1[i] == str2[i]:  
 matchCnt += 1  
  
print(**"Similarity between two said strings:"**)  
print(matchCnt / long)

**Output:**

Enter String 1

Python Exercises

Enter String 2

Python Exercises

Similarity between two said strings:

1.0

Enter String 1

Python Exercises

Enter String 2

Python Exercise

Similarity between two said strings:

0.9375

Q4) Aim: Discuss different collections like list, tuple and dictionary

1. Write a python program to implement insertion sort and merge sort using lists

import random

def merge\_sort(lst):  
 if len(lst) > 1:  
 mid = len(lst) // 2  
 left\_half = lst[:mid]  
 right\_half = lst[mid:]  
  
 merge\_sort(left\_half)  
 merge\_sort(right\_half)  
  
 i = j = k = 0  
 while i < len(left\_half) and j < len(right\_half):  
 if left\_half[i] < right\_half[j]:  
 lst[k] = left\_half[i]  
 i += 1  
 else:  
 lst[k] = right\_half[j]  
 j += 1  
 k += 1  
  
 while i < len(left\_half):  
 lst[k] = left\_half[i]  
 i += 1  
 k += 1  
  
 while j < len(right\_half):  
 lst[k] = right\_half[j]  
 j += 1  
 k += 1  
  
 return lst  
  
def insertion\_sort(arr):  
 for i in range(1, len(arr)):  
 key = arr[i]  
 j = i - 1  
 while j >= 0 and key < arr[j]:  
 arr[j + 1] = arr[j]  
 j -= 1  
 arr[j + 1] = key  
  
my\_list = []  
for i in range(10):  
 my\_list.append(random.randint(0, 999))  
  
print("\nUnsorted List")  
print(my\_list)  
print("Sorting using Insertion Sort")  
insertion\_sort(my\_list)  
print(my\_list)  
my\_list = []  
for i in range(10):  
 my\_list.append(random.randint(0, 999))  
  
print("\nUnsorted List")  
print(my\_list)  
print("Sorting using Merge Sort")  
merge\_sort(my\_list)  
print(my\_list)

**Output:**

Unsorted List

[932, 111, 226, 685, 543, 589, 918, 539, 294, 717]

Sorting using Insertion Sort

[111, 226, 294, 539, 543, 589, 685, 717, 918, 932]

Unsorted List

[613, 176, 828, 265, 65, 326, 359, 919, 514, 868]

Sorting using Merge Sort

[65, 176, 265, 326, 359, 514, 613, 828, 868, 919]

1. Write a program to convert roman numbers in to integer values using dictionaries.

def roman2Dec(romStr):  
 roman\_dict ={'I': 1, 'V': 5, 'X': 10, 'L': 50, 'C': 100, 'D': 500, 'M': 1000}  
 *# Analyze string backwards* romanBack = list(romStr)[::-1]  
 value = 0  
 *# To keep track of order* rightVal = roman\_dict[romanBack[0]]   
 for numeral in romanBack:  
 leftVal = roman\_dict[numeral]  
 *# Check for subtraction* if leftVal < rightVal:  
 value -= leftVal  
 else:  
 value += leftVal  
 rightVal = leftVal  
 return value  
  
romanStr = input("Enter a Roman Number : ")  
print(roman2Dec(romanStr))

**Output:**

Enter a Roman Number : XVII

17

Enter a Roman Number : MLXVI

1066

Q5) Aim: Demonstration of pattern recognition with and without using regular expressions

1. Write a function called isphonenumber () to recognize a pattern 415-555-4242 without using regular expression and also write the code to recognize the same pattern using regular expression.

import re  
  
def isphonenumber(numStr):  
 if len(numStr) != 12:  
 return False  
 for i in range(len(numStr)):  
 if i==3 or i==7:  
 if numStr[i] != "-":  
 return False  
 else:  
 if numStr[i].isdigit() == False:  
 return False  
 return True  
  
  
def chkphonenumber(numStr):  
 ph\_no\_pattern = re.compile(r'^\d{3}-\d{3}-\d{4}$')  
 if ph\_no\_pattern.match(numStr):  
 return True  
 else:  
 return False  
  
ph\_num = input("Enter a phone number : ")  
print("Without using Regular Expression")  
if isphonenumber(ph\_num):  
 print("Valid phone number")  
else:  
 print("Invalid phone number")  
  
print("Using Regular Expression")  
if chkphonenumber(ph\_num):  
 print("Valid phone number")  
else:  
 print("Invalid phone number")

**Output:**

Enter a phone number : 444-654-5656

Without using Regular Expression

Valid phone number

Using Regular Expression

Valid phone number

Enter a phone number : 45A4-444-878

Without using Regular Expression

Invalid phone number

Using Regular Expression

Invalid phone number

1. Develop a python program that could search the text in a file for phone numbers

(+919900889977) and email addresses ([sample@gmail.com](mailto:sample@gmail.com))

import re

*# Define the regular expression for phone numbers*phone\_regex = re.compile(r'\+\d{12}')  
email\_regex = re.compile(r'[A-Za-z0-9.\_]+@[A-Za-z0-9]+\.[A-Z|a-z]{2,}')  
*# Open the file for reading*with open('example.txt', 'r') as f:  
 *# Loop through each line in the file* for line in f:  
 *# Search for phone numbers in the line* matches = phone\_regex.findall(line)  
 *# Print any matches found* for match in matches:  
 print(match)  
  
 matches = email\_regex.findall(line)  
 *# Print any matches found* for match in matches:  
 print(match)

**Output:**

+918151894220

+829392938876

+918768456234

mitstore@gmail.in

Q6) Aim: Demonstration of reading, writing and organizing files.

1. Write a python program to accept a file name from the user and perform the following operations
   1. Display the first N line of the file
   2. Find the frequency of occurrence of the word accepted from the user in the file

import os.path  
import sys  
fname = input("Enter the filename : ")  
  
if not os.path.isfile(fname):  
 print("File", fname, "doesn't exists")  
 sys.exit(0)  
  
infile = open(fname, "r")  
  
lineList = infile.readlines()  
  
for i in range(20):  
 print(i + 1, ":", lineList[i])  
  
word = input("Enter a word : ")  
cnt = 0  
for line in lineList:  
 cnt += line.count(word)  
  
print("The word", word, "appears", cnt, "times in the file")

**Output:**

Enter the filename : example.txt

1 : this is phone number +918151894220

2 : no phone number here

3 : here we have one +829392938876

4 : we have an email prakash81.82@gmail.in and a number +918768456234

5 : nothing of that sort here

6 : Better hope the life-inspector doesn't come around while you have your

7 : life in such a mess.

8 : You can create your own opportunities this week. Blackmail a senior executive.

9 : Be different: conform.

10 : Be cheerful while you are alive.

11 : -- Phathotep, 24th Century B.C.

12 : Q: How many journalists does it take to screw in a light bulb?

13 : A: Three. One to report it as an inspired government program to bring

14 : light to the people, one to report it as a diabolical government plot

15 : to deprive the poor of darkness, and one to win a Pulitzer prize for

16 : reporting that Electric Company hired a light bulb-assassin to break

17 : the bulb in the first place.

18 : Q: Why did the astrophysicist order three hamburgers?

19 : A: Because he was hungry.

20 : Q: Why haven't you graduated yet?

Enter a word : the

The word the appears 7 times in the file

1. Write a python program to create a ZIP file of a particular folder which contains several files inside it.

import os  
import sys  
import pathlib  
import zipfile  
  
dirName = input("Enter Directory name that you want to backup : ")  
  
if not os.path.isdir(dirName):  
 print("Directory", dirName, "doesn't exists")  
 sys.exit(0)  
  
curDirectory = pathlib.Path(dirName)  
  
with zipfile.ZipFile("myZip.zip", mode="w") as archive:  
 for file\_path in curDirectory.rglob("\*"):  
 archive.write(file\_path, arcname=file\_path.relative\_to(curDirectory))  
  
if os.path.isfile("myZip.zip"):  
 print("Archive", "myZip.zip", "created successfully")  
else:  
 print("Error in creating zip archive")

**Output:**

Enter Directory name that you want to backup : zipDemo

Archive myZip.zip created successfully

Q7) Aim: Demonstration of the concepts of classes, methods, objects and inheritance.

1. By using the concept of inheritance write a python program to find the area of triangle, circle and rectangle.

import math  
class Shape:  
 def \_\_init\_\_(self):  
 self.area = 0  
 self.name = ""  
 def showArea(self):  
 print("The area of the", self.name, "is", self.area, "units")  
  
class Circle(Shape):  
 def \_\_init\_\_(self, radius):  
 self.area = 0  
 self.name = "Circle"  
 self.radius = radius  
  
 def calcArea(self):  
 self.area = math.pi \* self.radius \* self.radius  
  
class Rectangle(Shape):  
 def \_\_init\_\_(self, length, breadth):  
 self.area = 0  
 self.name = "Rectangle"  
 self.length = length  
 self.breadth = breadth  
  
 def calcArea(self):  
 self.area = self.length \* self.breadth  
  
class Triangle(Shape):  
 def \_\_init\_\_(self, base, height):  
 self.area = 0  
 self.name = "Triangle"  
 self.base = base  
 self.height = height  
  
 def calcArea(self):  
 self.area = self.base \* self.height / 2  
  
c1 = Circle(5)  
c1.calcArea()  
c1.showArea()  
r1 = Rectangle(5, 4)  
r1.calcArea()  
r1.showArea()  
t1 = Triangle(3, 4)  
t1.calcArea()  
t1.showArea()

**Output:**

The area of the Circle is 78.53981633974483 units

The area of the Rectangle is 20 units

The area of the Triangle is 6.0 units

1. Write a python program by creating a class called Employee to store the details of Name, Employee\_ID, Department and Salary, and implement a method to update salary of employees belonging to a given department.

**class** Employee:  
 **def** \_\_init\_\_(self):  
 self.name = **""** self.empId = **""** self.dept = **""** self.salary = 0  
  
 **def** getEmpDetails(self):  
 self.name = input(**"Enter Employee name : "**)  
 self.empId = input(**"Enter Employee ID : "**)  
 self.dept = input(**"Enter Employee Dept : "**)  
 self.salary = int(input(**"Enter Employee Salary : "**))  
  
 **def** showEmpDetails(self):  
 print(**"Employee Details"**)  
 print(**"Name : "**, self.name)  
 print(**"ID : "**, self.empId)  
 print(**"Dept : "**, self.dept)  
 print(**"Salary : "**, self.salary)  
  
 **def** updtSalary(self):  
 self.salary = int(input(**"Enter new Salary : "**))  
 print(**"Updated Salary"**, self.salary)  
  
  
e1 = Employee()  
e1.getEmpDetails()  
e1.showEmpDetails()  
e1.updtSalary()

**Output:**Enter Employee name : Samantha

Enter Employee ID : A123

Enter Employee Dept : ISE

Enter Employee Salary : 85750

Employee Details

Name : Samantha

ID : A123

Dept : ISE

Salary : 85750

Enter new Salary : 88800

Updated Salary 88800

Q8) Aim: Demonstration of classes and methods with polymorphism and overriding

1. Write a python program to find the whether the given input is palindrome or not (for both string and integer) using the concept of polymorphism and inheritance.

class PaliStr:  
 def \_\_init\_\_(self):  
 self.isPali = False  
  
 def chkPalindrome(self, myStr):  
 if myStr == myStr[::-1]:  
 self.isPali = True  
 else:  
 self.isPali = False  
  
 return self.isPali  
  
class PaliInt(PaliStr):  
 def \_\_init\_\_(self):  
 self.isPali = False  
  
 def chkPalindrome(self, val):  
 temp = val  
 rev = 0  
 while temp != 0:  
 dig = temp % 10  
 rev = (rev \* 10) + dig  
 temp = temp // 10  
  
 if val == rev:  
 self.isPali = True  
 else:  
 self.isPali = False  
  
 return self.isPali  
  
st = input("Enter a string : ")  
stObj = PaliStr()  
if stObj.chkPalindrome(st):  
 print("Given string is a Palindrome")  
else:  
 print("Given string is not a Palindrome")  
  
val = int(input("Enter a integer : "))  
  
intObj = PaliInt()  
if intObj.chkPalindrome(val):  
 print("Given integer is a Palindrome")  
else:  
 print("Given integer is not a Palindrome")

**Output:**

Enter a string : madam

Given string is a Palindrome

Enter a integer : 567587

Given integer is not a Palindrome

Enter a string : INDIA

Given string is not a Palindrome

Enter a integer : 6789876

Given integer is a Palindrome

Q9) Aim: Demonstration of working with excel spreadsheets and web scraping

1. Write a python program to download the all XKCD comics

import requests  
import os  
from bs4 import BeautifulSoup  
  
*# Set the URL of the first XKCD comic*url = 'https://xkcd.com/1/'  
  
*# Create a folder to store the comics*if not os.path.exists('xkcd\_comics'):  
 os.makedirs('xkcd\_comics')  
  
*# Loop through all the comics*while True:  
 *# Download the page content* res = requests.get(url)  
 res.raise\_for\_status()  
  
 *# Parse the page content using BeautifulSoup* soup = BeautifulSoup(res.text, 'html.parser')  
  
 *# Find the URL of the comic image* comic\_elem = soup.select('#comic img')  
 if comic\_elem == []:  
 print('Could not find comic image.')  
 else:  
 comic\_url = 'https:' + comic\_elem[0].get('src')  
  
 *# Download the comic image* print(f'Downloading {comic\_url}...')  
 res = requests.get(comic\_url)  
 res.raise\_for\_status()  
  
 *# Save the comic image to the xkcd\_comics folder* image\_file = open(os.path.join('xkcd\_comics', os.path.basename(comic\_url)), 'wb')  
 for chunk in res.iter\_content(100000):  
 image\_file.write(chunk)  
 image\_file.close()  
  
 *# Get the URL of the previous comic* prev\_link = soup.select('a[rel="prev"]')[0]  
 if not prev\_link:  
 break  
 url = 'https://xkcd.com' + prev\_link.get('href')  
  
print('All comics downloaded.')

**Output:**

Downloading https://imgs.xkcd.com/comics/barrel\_cropped\_(1).jpg...

Downloading https://imgs.xkcd.com/comics/radians\_are\_cursed.png...

Downloading https://imgs.xkcd.com/comics/presents\_for\_biologists.png...

Downloading https://imgs.xkcd.com/comics/launch\_window.png...

Downloading https://imgs.xkcd.com/comics/obituary\_editor.png...

Downloading https://imgs.xkcd.com/comics/fanservice.png...

Downloading https://imgs.xkcd.com/comics/hand\_dryers.png...

1. Demonstrate python program to read the data from the spreadsheet and write the data in to the spreadsheet

from openpyxl import Workbook  
from openpyxl.styles import Font  
  
wb = Workbook()  
sheet = wb.active  
sheet.title = "Language"  
wb.create\_sheet(title="Capital")  
  
lang = ["Kannada", "Telugu", "Tamil"]  
state = ["Karnataka", "Telangana", "Tamil Nadu"]  
capital = ["Bengaluru", "Hyderabad", "Chennai"]  
code = ['KA', 'TS', 'TN']  
  
sheet.cell(row=1, column=1).value = "State"  
sheet.cell(row=1, column=2).value = "Language"  
sheet.cell(row=1, column=3).value = "Code"  
  
ft = Font(bold=True)  
for row in sheet["A1:C1"]:  
 for cell in row:  
 cell.font = ft  
  
for i in range(2, 5):  
 sheet.cell(row=i, column=1).value = state[i - 2]  
 sheet.cell(row=i, column=2).value = lang[i - 2]  
 sheet.cell(row=i, column=3).value = code[i - 2]  
  
wb.save("demo.xlsx")  
  
sheet = wb["Capital"]  
  
sheet.cell(row=1, column=1).value = "State"  
sheet.cell(row=1, column=2).value = "Capital"  
sheet.cell(row=1, column=3).value = "Code"  
  
ft = Font(bold=True)  
for row in sheet["A1:C1"]:  
 for cell in row:  
 cell.font = ft  
  
for i in range(2, 5):  
 sheet.cell(row=i, column=1).value = state[i - 2]  
 sheet.cell(row=i, column=2).value = capital[i - 2]  
 sheet.cell(row=i, column=3).value = code[i - 2]  
  
wb.save("demo.xlsx")  
  
srchCode = input("Enter state code for finding capital ")  
for i in range(2, 5):  
 data = sheet.cell(row=i, column=3).value  
 if data == srchCode:  
 print("Corresponding capital for code", srchCode, "is", sheet.cell(row=i, column=2).value)  
  
sheet = wb["Language"]  
  
srchCode = input("Enter state code for finding language ")  
for i in range(2, 5):  
 data = sheet.cell(row=i, column=3).value  
 if data == srchCode:  
 print("Corresponding language for code", srchCode, "is", sheet.cell(row=i, column=2).value)  
  
wb.close()

**Output:**

Enter state code for finding capital KA

Corresponding capital for code KA is Bengaluru

Enter state code for finding language TS

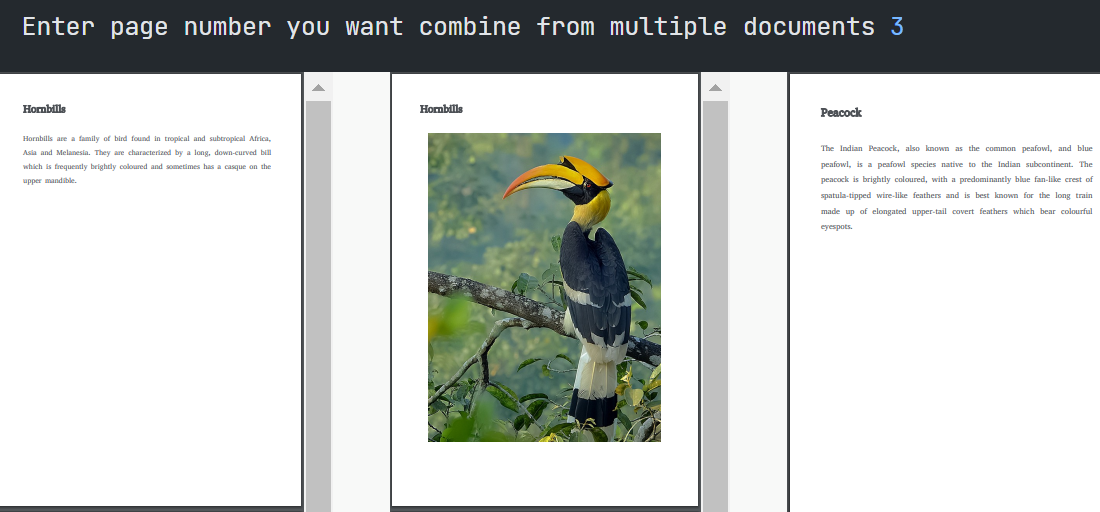
Corresponding language for code TS is Telugu

Q10) Aim: Demonstration of working with PDF, word and JSON files

1. Write a python program to combine select pages from many PDFs

from PyPDF2 import PdfWriter, PdfReader  
  
num = int(input("Enter page number you want combine from multiple documents "))  
  
pdf1 = open('birds.pdf', 'rb')  
pdf2 = open('birdspic.pdf', 'rb')  
  
pdf\_writer = PdfWriter()  
  
pdf1\_reader = PdfReader(pdf1)  
page = pdf1\_reader.pages[num - 1]  
pdf\_writer.add\_page(page)  
  
pdf2\_reader = PdfReader(pdf2)  
page = pdf2\_reader.pages[num - 1]  
pdf\_writer.add\_page(page)  
  
with open('output.pdf', 'wb') as output:  
 pdf\_writer.write(output)

**Output:**

****

1. Write a python program to fetch current weather data from the JSON file

import json  
  
*# Load the JSON data from file*with open('weather\_data.json') as f:  
 data = json.load(f)  
  
*# Extract the required weather data*current\_temp = data['main']['temp']  
humidity = data['main']['humidity']  
weather\_desc = data['weather'][0]['description']  
  
*# Display the weather data*print(f"Current temperature: {current\_temp}°C")  
print(f"Humidity: {humidity}%")  
print(f"Weather description: {weather\_desc}")

**Output:**

Current temperature: 15.45°C

Humidity: 64%

Weather description: clear sky