Python Programming for Data Handling Practical File

Name: ANSH RAJ

Examination Roll No.: 22003582112

Roll No.: 22/25081

Semester: III

1. Write a Python program to calculate the factorial of a number.

def factorial(n):

if n == 0 or n == 1:

return 1

else:

return n \* factorial(n - 1)

# Get user input

num = int(input("Enter a number: "))

# Check if the number is negative

if num < 0:

print("Factorial is not defined for negative numbers.")

else:

result = factorial(num)

print(f"The factorial of {num} is: {result}")

output:-

Enter a number: 5

The factorial of 5 is: 120

2. Write a Python program to generate prime numbers between 1 to n, where n is provided as input by the user.

def is\_prime(num):

if num < 2:

return False

for i in range(2, int(num\*\*0.5) + 1):

if num % i == 0:

return False

return True

def generate\_primes(n):

primes = [num for num in range(2, n+1) if is\_prime(num)]

return primes

n = int(input("Enter a value of n: "))

if n < 0:

print("Please enter a non-negative number.")

else:

prime\_numbers = generate\_primes(n)

print(f"Prime numbers between 1 and {n}: {prime\_numbers}")

output:-

Enter a value of n: 19

Prime numbers between 1 and 19: [2, 3, 5, 7, 11, 13, 17, 19]

3. Write a Python program to find the sum and average of numbers in a given list.

def calculate\_sum\_and\_average(numbers):

total = sum(numbers)

average = total / len(numbers) if len(numbers) > 0 else 0

return total, average

input\_numbers = input("Enter a list of numbers separated by spaces: ")

numbers = [float(x) for x in input\_numbers.split() if x.replace(".", "", 1).isdigit()]

if not numbers:

print("Please enter at least one valid number.")

else:

sum\_result, average\_result = calculate\_sum\_and\_average(numbers)

print(f"Sum: {sum\_result}")

print(f"Average: {average\_result}")

output:-

Enter a list of numbers separated by spaces: 4 6 7 3 5

Sum: 25.0

Average: 5.0

4) Given two sets, set1 and set2, write a Python program to find their union, intersection, and difference.

set1 = {1, 2, 3, 4, 5}

set2 = {3, 4, 5, 6, 7}

union\_set = set1.union(set2)

print(f"Union of set1 and set2: {union\_set}")

intersection\_set = set1.intersection(set2)

print(f"Intersection of set1 and set2: {intersection\_set}")

difference\_set1 = set1.difference(set2)

print(f"Difference of set1 and set2: {difference\_set1}")

difference\_set2 = set2.difference(set1)

print(f"Difference of set2 and set1: {difference\_set2}")

output:-

Union of set1 and set2: {1, 2, 3, 4, 5, 6, 7}

Intersection of set1 and set2: {3, 4, 5}

Difference of set1 and set2: {1, 2}

Difference of set2 and set1: {6, 7}

5) Given a list of numbers, write a Python program to count the number of times an element occurs in a list and create a dictionary with element:count as key:value pairs.

def count\_elements(lst):

element\_count\_dict = {}

for element in lst:

if element in element\_count\_dict:

element\_count\_dict[element] += 1

else:

element\_count\_dict[element] = 1

return element\_count\_dict

numbers\_list = [1, 2, 3, 1, 2, 4, 1, 3, 5]

element\_count\_dictionary = count\_elements(numbers\_list)

print("Original List:", numbers\_list)

print("Element Count Dictionary:", element\_count\_dictionary)

output:-

Original List: [1, 2, 3, 1, 2, 4, 1, 3, 5]

Element Count Dictionary: {1: 3, 2: 2, 3: 2, 4: 1, 5: 1}

6) Write a Python program to swap the first two and last two characters of a given string.

def swap\_first\_and\_last\_two\_chars(input\_str):

if len(input\_str) < 2:

return input\_str

first\_two\_chars = input\_str[:2]

last\_two\_chars = input\_str[-2:]

middle\_chars = input\_str[2:-2]

swapped\_str = last\_two\_chars + middle\_chars + first\_two\_chars

return swapped\_str

input\_string = input("Enter a string: ")

result\_string = swap\_first\_and\_last\_two\_chars(input\_string)

print("Original String:", input\_string)

print("Swapped String:", result\_string)

output:-

Enter a string: python language

Original String: python language

Swapped String: gethon languapy

7) Write a Python program to create a text file having names of ten Indian cities.

cities = ["Mumbai", "Delhi", "Bangalore", "Hyderabad", "Chennai", "Kolkata", "Ahmedabad", "Pune", "Jaipur", "Lucknow"]

file\_name = "indian\_cities.txt"

with open(file\_name, 'w') as file:

for city in cities:

file.write(city + '\n')

print(f"The file '{file\_name}' has been created with the names of ten Indian cities.")

output:-

The file 'indian\_cities.txt' has been created with the names of ten Indian cities.

8) Write a Python program to create a text file having atleast five lines about your college using

writelines() function.

file\_name = "college\_description.txt"

college\_description = [

"Welcome to ARSD College!",

" ARSD College is known for its outstanding academic programs.",

"Our campus is equipped with state-of-the-art facilities and a vibrant student community.",

"Dedicated faculty members strive for excellence in teaching and research.",

"Join us at ARSD College and embark on a journey of knowledge and personal growth."

]

with open(file\_name, 'w') as file:

file.writelines(line + '\n' for line in college\_description)

print(f"The file '{file\_name}' has been created with a description of the college.")

output:-

The file 'college\_description.txt' has been created with a description of the college.

9) Write a Python program which reads the data from three input files having Employee Names and merges them into one output file.

input\_files = ["input1.txt", "input2.txt", "input3.txt"]

output\_file = "merged\_output.txt"

def merge\_files(input\_files, output\_file):

with open(output\_file, 'w') as output:

for input\_file in input\_files:

with open(input\_file, 'r') as input\_data:

output.write(input\_data.read())

output.write('\n') # Add a newline between data from different files

merge\_files(input\_files, output\_file)

print(f"The data from {len(input\_files)} input files has been merged into '{output\_file}'.")

output:-

The data from 3 input files has been merged into 'merged\_output.txt'.

10) Write a Python program to count the number of vowels in a file and write the vowel: count in a dictionary.

def count\_vowels(file\_path):

vowels = "aeiouAEIOU"

vowel\_count = {vowel: 0 for vowel in vowels}

try:

with open(file\_path, 'r') as file:

content = file.read()

for char in content:

if char in vowels:

vowel\_count[char] += 1

except FileNotFoundError:

print(f"File '{file\_path}' not found.")

return None

return vowel\_count

file\_path = "vowel.txt"

vowel\_count\_dict = count\_vowels(file\_path)

if vowel\_count\_dict is not None:

print("Vowel Count Dictionary:")

for vowel, count in vowel\_count\_dict.items():

print(f"{vowel}: {count}")

output:-

Vowel Count Dictionary:

a: 8

e: 2

i: 0

o: 2

u: 0

A: 1

E: 0

I: 1

O: 0

U: 0

11) Write a Python program to create a CSV file having student data: Roll\_No, Enrollment No, Name,Course, Semester.

import csv

students\_data = [

{"Roll\_No": 101, "Enrollment\_No": "EN12345", "Name": "John Doe", "Course": "Computer Science", "Semester": 3},

{"Roll\_No": 102, "Enrollment\_No": "EN67890", "Name": "Jane Smith", "Course": "Electrical Engineering", "Semester": 2},

{"Roll\_No": 103, "Enrollment\_No": "EN54321", "Name": "Alice Johnson", "Course": "Mechanical Engineering", "Semester": 4},

{"Roll\_No": 104, "Enrollment\_No": "EN98765", "Name": "Bob Williams", "Course": "Civil Engineering", "Semester": 1},

]

csv\_file\_name = "student\_data.csv"

with open(csv\_file\_name, mode='w', newline='') as csv\_file:

fieldnames = ["Roll\_No", "Enrollment\_No", "Name", "Course", "Semester"]

writer = csv.DictWriter(csv\_file, fieldnames=fieldnames)

writer.writeheader()

writer.writerows(students\_data)

print(f"The CSV file '{csv\_file\_name}' has been created with student data.")

output:-

The CSV file 'student\_data.csv' has been created with student data.

12) Write a Python program library to read the CSV file created in the above program and filter out records of II semester students.

import csv

def filter\_semester\_records(csv\_file\_path, target\_semester):

filtered\_records = []

with open(csv\_file\_path, 'r') as csv\_file:

reader = csv.DictReader(csv\_file)

for row in reader:

if int(row["Semester"]) == target\_semester:

filtered\_records.append(row)

return filtered\_records

csv\_file\_path = "student\_data.csv"

target\_semester = 2

filtered\_records = filter\_semester\_records(csv\_file\_path, target\_semester)

if filtered\_records:

print(f"Records for {target\_semester} semester:")

for record in filtered\_records:

print(record)

else:

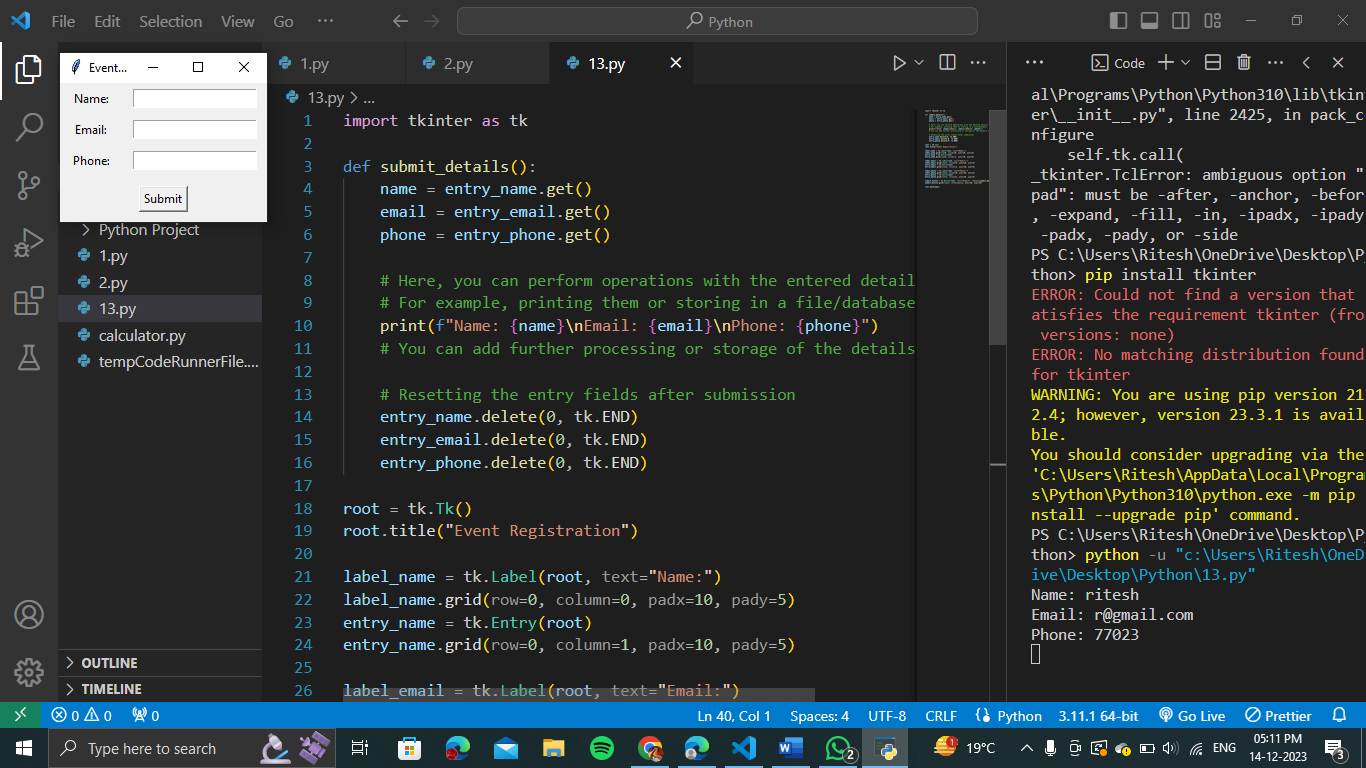
print(f"No records found for {target\_semester} semester.")

output:-

Records for 2 semester:

{'Roll\_No': '102', 'Enrollment\_No': 'EN67890', 'Name': 'Jane Smith', 'Course': 'Electrical Engineering', 'Semester': '2'}

13. Write a Python program using tkinter library to create a GUI to enter registration details for an event.



14. Write a Python program using tkinter library to create a calculator to perform addition, subtraction, multiplication and division of two numbers entered by the user.

import tkinter as tk

def calculate():

    try:

        num1 = float(entry\_num1.get())

        num2 = float(entry\_num2.get())

        operation = operation\_var.get()

        if operation == "Addition":

            result = num1 + num2

        elif operation == "Subtraction":

            result = num1 - num2

        elif operation == "Multiplication":

            result = num1 \* num2

        elif operation == "Division":

            if num2 == 0:

                result = "Cannot divide by zero"

            else:

                result = num1 / num2

        else:

            result = "Invalid operation"

        result\_label.config(text="Result: " + str(result))

    except ValueError:

        result\_label.config(text="Please enter valid numbers")

root = tk.Tk()

root.title("Simple Calculator")

label\_num1 = tk.Label(root, text="Enter number 1:")

label\_num1.grid(row=0, column=0, padx=10, pady=5)

entry\_num1 = tk.Entry(root)

entry\_num1.grid(row=0, column=1, padx=10, pady=5)

label\_num2 = tk.Label(root, text="Enter number 2:")

label\_num2.grid(row=1, column=0, padx=10, pady=5)

entry\_num2 = tk.Entry(root)

entry\_num2.grid(row=1, column=1, padx=10, pady=5)

operation\_var = tk.StringVar()

operation\_var.set("Addition")

operation\_label = tk.Label(root, text="Choose operation:")

operation\_label.grid(row=2, column=0, padx=10, pady=5)

operation\_menu = tk.OptionMenu(root, operation\_var, "Addition", "Subtraction", "Multiplication", "Division")

operation\_menu.grid(row=2, column=1, padx=10, pady=5)

calculate\_button = tk.Button(root, text="Calculate", command=calculate)

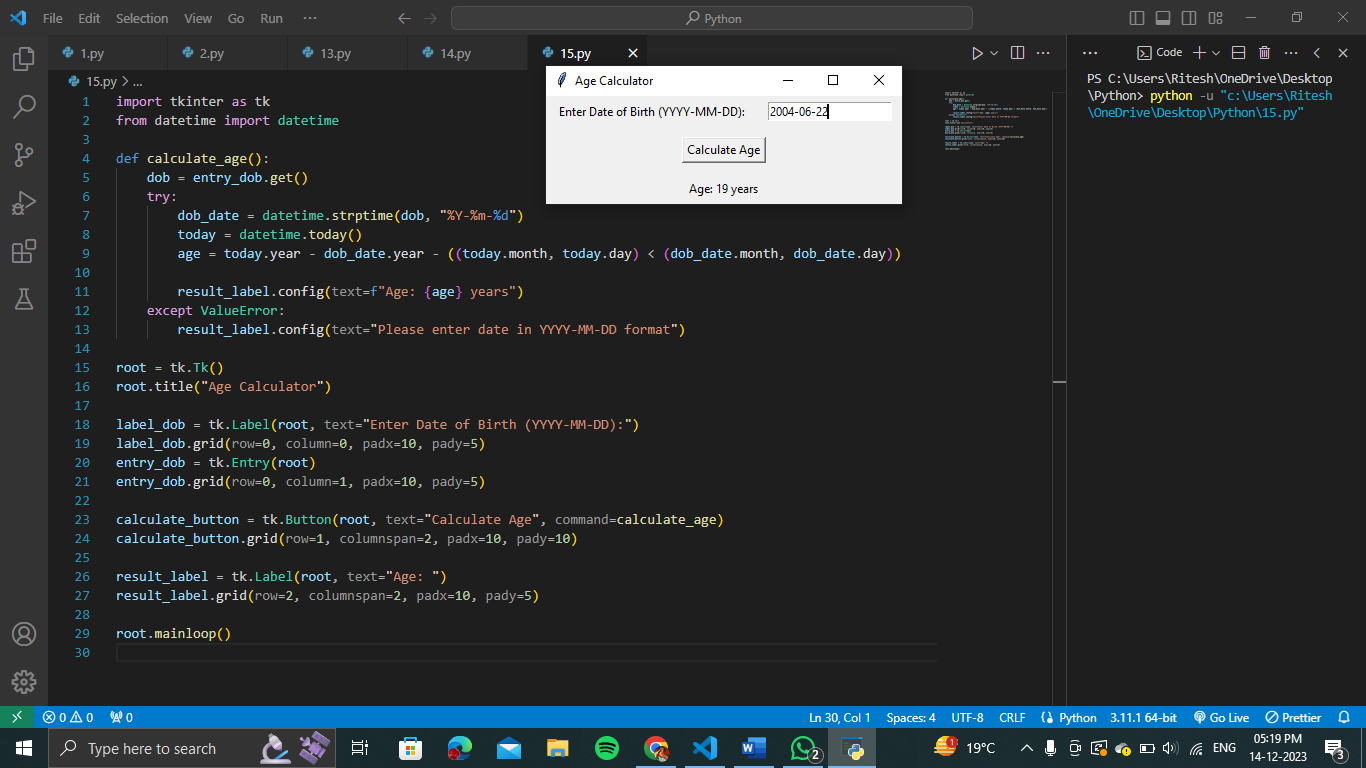
calculate\_button.grid(row=3, columnspan=2, padx=10, pady=10)

result\_label = tk.Label(root, text="Result: ")

result\_label.grid(row=4, columnspan=2, padx=10, pady=5)

root.mainloop()

15. Write a Python program using tkinter library to create an age calculator to calculate age when DOB is entered.



16. Write a Python program using tkinter library to read and write student details namely Roll\_No, Enrollment\_No, Name, course, Semester through a form and write the entered details to a CSV file.

import tkinter as tk

import csv

# Define student details

fields = ["Roll\_No", "Enrollment\_No", "Name", "Course", "Semester"]

# Function to read data from CSV file

def read\_data():

# Open file in read mode

with open("students.csv", "r") as csvfile:

reader = csv.reader(csvfile)

# Skip header (first line)

next(reader)

# Store data in a list of dictionaries

data = [dict(zip(fields, row)) for row in reader]

return data

# Function to write data to CSV file

def write\_data(data):

# Open file in write mode

with open("students.csv", "w", newline="") as csvfile:

writer = csv.DictWriter(csvfile, fieldnames=fields)

# Write header row

writer.writeheader()

# Write data rows

for student in data:

writer.writerow(student)

# Function to handle form submission

def submit\_click():

# Get data from entry fields

data = {}

for field, entry in entries.items():

data[field] = entry.get()

# Read existing data from CSV

existing\_data = read\_data()

# Update data (add new entry)

existing\_data.append(data)

# Write updated data to CSV

write\_data(existing\_data)

# Reset form

for entry in entries.values():

entry.delete(0, tk.END)

# Show success message

success\_label.config(text="Student details saved successfully!")

# Initialize main window

root = tk.Tk()

root.title("Student Registration")

# Create frame for form

frame = tk.Frame(root)

frame.pack(pad=10)

# Create labels and entry fields for each detail

entries = {}

for field in fields:

field\_label = tk.Label(frame, text=field)

field\_label.pack(pady=5)

field\_entry = tk.Entry(frame)

entries[field] = field\_entry

field\_entry.pack()

# Create button to submit form

submit\_button = tk.Button(frame, text="Submit", command=submit\_click)

submit\_button.pack(pady=10)

# Create label to show success message

success\_label = tk.Label(frame, text="")

success\_label.pack(pady=5)

# Start the main loop

root.mainloop()