**Weather modeling using the quadratic solution in stages:**

Hard-coding variables (keyboard input):

import matplotlib.pyplot as plt

def quadratic\_temperature\_model(day, a, b, c):

    temperature = a \* day\*\*2 + b \* day + c  # Quadratic equation for temperature

    return temperature

def main():

    # Fixed coefficients for the quadratic equation: ax^2 + bx + c

    a, b, c = 2, 3, 1

    # Fixed number of days to model

    num\_days = int(input("Enter the number of days to model: "))

    # Lists to store day and corresponding temperature values

    days = list(range(1, num\_days + 1))

    temperatures = [quadratic\_temperature\_model(day, a, b, c) for day in days]

    # Plotting

    plt.plot(days, temperatures, label=f'Temperature: {a}x^2 + {b}x + {c}')

    plt.title('Weather Modeling')

    plt.xlabel('Day')

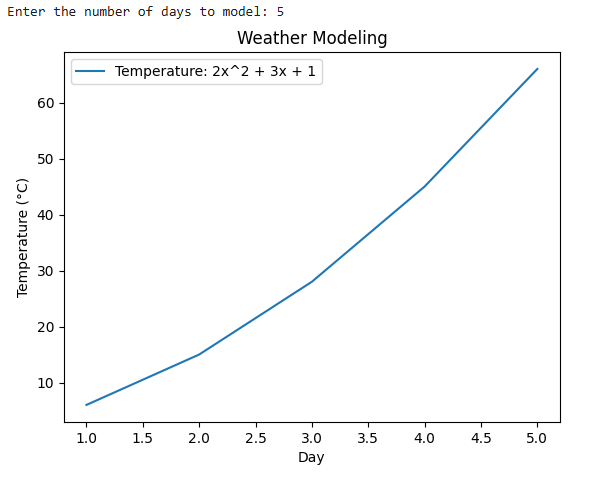
    plt.ylabel('Temperature (°C)')

    plt.legend()

    plt.show()

main()

OUTPUT:



Hard-coding variables (user input):

import matplotlib.pyplot as plt

def quadratic\_temperature\_model(day, a, b, c):

    temperature = a \* day\*\*2 + b \* day + c

    return temperature

def main():

    # Hard-coded coefficients for the quadratic equation: ax^2 + bx + c

    a = float(input("Enter the quadratic coefficient (a): "))

    b = float(input("Enter the linear coefficient (b): "))

    c = float(input("Enter the constant term (c): "))

    # Get user input for the number of days to model

    num\_days = int(input("Enter the number of days to model: "))

    days = list(range(1, num\_days + 1))

    temperatures = [quadratic\_temperature\_model(day, a, b, c) for day in days]

    # Plotting

    plt.plot(days, temperatures, marker='o', label=f'Temperature: {a}x^2 + {b}x + {c}')

    plt.title('Weather Modeling')

    plt.xlabel('Day')

    plt.ylabel('Temperature (°C)')

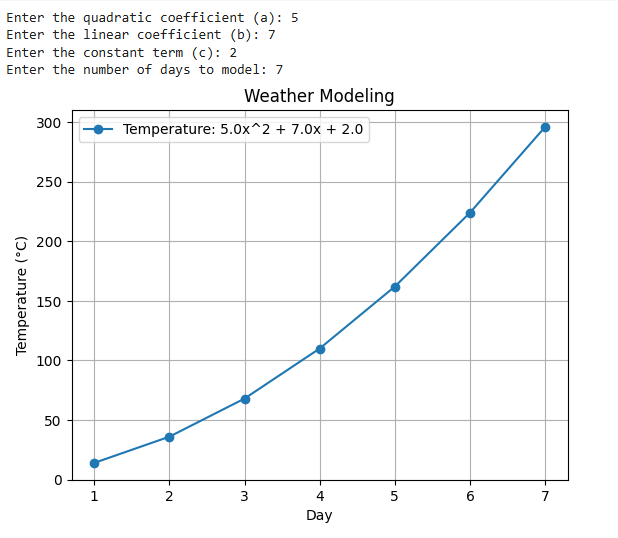
    plt.grid()

    plt.legend()

    plt.show()

main()

OUTPUT:



Read from a file (single set of inputs):

import matplotlib.pyplot as plt

import pandas as pd

def quadratic\_weather\_model(time, a, b, c):

    temperature = a \* (time \*\* 2) + b \* time + c

    return temperature

def main():

    print("Quadratic Weather Modeling")

    print("-------------------------")

    try:

        file\_path = 'wea2.csv'

        df = pd.read\_csv(file\_path)

        time\_values = list(range(0, 11))

        plt.figure(figsize=(8, 6))

        for index, row in df.iterrows():

            x, y, z = row['a'], row['b'], row['c']

            temperature\_values = [quadratic\_weather\_model(t, x, y, z) for t in time\_values]

            plt.plot(time\_values, temperature\_values, marker='o', linestyle='-', label=f'a={x}, b={y}, c={z}')

        plt.title('Temperature Variation Over Time')

        plt.xlabel('Time')

        plt.ylabel('Temperature')

        plt.grid(True)

        plt.xlim(0, 10)

        plt.legend()

        plt.show()

    except FileNotFoundError:

        print("File not found. Please make sure 'wea2.csv' exists.")

main()

OUTPUT:

Quadratic Weather Modeling

-------------------------

File not found. Please make sure 'wea2.csv' exists.

CSV FILE:

import pandas as pd

data = {

    'a': [1, 0.5],

    'b': [2, -1],

    'c': [10, 20]

}

df = pd.DataFrame(data)

df.to\_csv('wea2.csv', index=False)

print("wea2.csv created successfully.")

import matplotlib.pyplot as plt

import pandas as pd

def quadratic\_weather\_model(time, a, b, c):

    temperature = a \* (time \*\* 2) + b \* time + c

    return temperature

def main():

    print("Quadratic Weather Modeling")

    print("-------------------------")

    try:

        file\_path = 'wea2.csv'

        df = pd.read\_csv(file\_path)

        time\_values = list(range(0, 11))

        plt.figure(figsize=(8, 6))

        for index, row in df.iterrows():

            x, y, z = row['a'], row['b'], row['c']

            temperature\_values = [quadratic\_weather\_model(t, x, y, z) for t in time\_values]

            plt.plot(time\_values, temperature\_values, marker='o', linestyle='-', label=f'a={x}, b={y}, c={z}')

        plt.title('Temperature Variation Over Time')

        plt.xlabel('Time')

        plt.ylabel('Temperature')

        plt.grid(True)

        plt.xlim(0, 10)

        plt.legend()

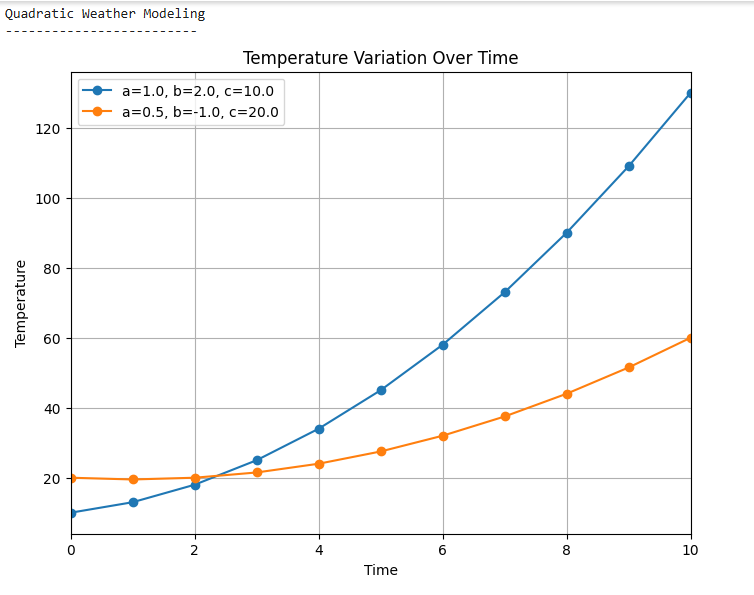
        plt.show()

    except FileNotFoundError:

        print("File not found. Please make sure 'wea2.csv' exists.")

main()

OUTPUT:



Read from a file (multiple sets of inputs):

CSV FILE:

import pandas as pd

# Pre-defined multiple sets of a, b, c

data = {

    'a': [0.5, 1.0, -0.2, 2.3],

    'b': [2.0, -3.0, 4.0, 0.5],

    'c': [20.0, 15.0, 10.0, 5.0]

}

# Create DataFrame

df = pd.DataFrame(data)

# Save to CSV

df.to\_csv('wea.csv', index=False)

print("✅ wea2.csv created successfully.")

import matplotlib.pyplot as plt

import pandas as pd

def quadratic\_weather\_model(time, a, b, c):

    temperature = a \* (time \*\* 2) + b \* time + c

    return temperature

def main():

    print("Quadratic Weather Modeling")

    print("-------------------------")

    try:

        file\_path = 'wea.csv'

        df = pd.read\_csv(file\_path)

        time\_values = list(range(0, 11))

        plt.figure(figsize=(8, 6))

        for index, row in df.iterrows():

            x, y, z = row['a'], row['b'], row['c']

            temperature\_values = [quadratic\_weather\_model(t, x, y, z) for t in time\_values]

            plt.plot(time\_values, temperature\_values, marker='o', linestyle='-', label=f'a={x}, b={y}, c={z}')

        plt.title('Temperature Variation Over Time')

        plt.xlabel('Time')

        plt.ylabel('Temperature')

        plt.grid(True)

        plt.xlim(0, 10)

        plt.legend()

        plt.show()

    except FileNotFoundError:

        print("File not found. Please make sure 'wea.csv' exists.")

main()

OUTPUT:

