```python
# -*- coding: utf-8 -*-
"""
Created on Fri Mar  7 10:49:50 2025

@author: PC
"""

"""
Robert Cocker
DATA 3461
Dr. Jawad
3/7/2025
Quiz #9 - Example 1
"""

# Example 1

# Import required libraries
import numpy as np
import matplotlib.pyplot as plt

# Dataset
x = np.array([2, 4, 5])
y = np.array([1.2, 2.8, 5.3])

# Plot
plt.scatter(x, y)

# Initialize parameters
b0 = 0 # intercept
b1 = 1 # slope
lr = 0.01 # learning rate
iterations = 1 # Number of iterations

for i in range(iterations):
    partial_wrt_b0 = b0
    partial_wrt_b1 = b1

    for j in range(len(x)):
        y_pred = b0 + b1 * x[j] # Predict value for given x

        error_cost = y_pred - y[j] # Calculate the error in prediction for all 3 poins

        partial_wrt_b0 += error_cost # Partial derivative 1
        partial_wrt_b1 += error_cost * x[j] # Partial derivative 2

    b0 -= lr * (2/len(x)) * partial_wrt_b0 # Update values
    b1 -= lr * (2/len(x)) * partial_wrt_b1 # Update values

# Value of coefficient 1:
print("b0 =", b0)

# Value of coefficent 2:
print("b1 =", b1)

# Predict new values:
y_pred = b0 + b1 * x
```

```
# Plot Regression line:
plt.plot(x, y_pred, color='red', label='Regression Line')
plt.legend()
plt.show()

# Questions - Inference
"""
Note: See run 1 submission for results (outputs)

Lecture (class) b0, b1 values and equation:
b0 = -0.0113
b1 = 0.9677
y-hat = 0.9677*x_i - 0.0113

a.) Ran code and observed results (output). Our calculations in Python match the handwritten methoc
    b0 and b1 values confirm our handwritten calculations though Python is more precise with 2> dec
    placement in calculations.

b.) Ran 100 iterations

c.) Ran 1000 iterations. Results: After 1000 iterations: b0 = -1.3902, b1 = 1.2306. This suggests t
    parameters compared to our initial calculations during lecture. Initial calculations: After 1 i
"""


"""
References
- lab, lecture, course materials
- ChatGPT
- Google
"""
```