**Q:5 Generate a model for Covid 19 with symptoms of parameters like fever, cold, shivering,**

**weight loss, generate 100 model data with random values for each parameter and order by**

**parameter lowest to highest in display based on the input parameter.**

Solution: COVID-19 Symptoms Model

Here’s a Python model that generates 100 random data points for COVID-19 symptoms (fever, cold, shivering, weight loss) and sorts the data by any specified symptom.

Code:

import random

import pandas as pd

class CovidSymptoms:

def \_init\_(self):

self.data = []

def generate\_data(self, num\_records=100):

for i in range(num\_records):

symptoms = {

"Fever": random.uniform(98.0, 104.0),

"Cold": random.randint(0, 1), # 0: No, 1: Yes

"Shivering": random.randint(0, 1), # 0: No, 1: Yes

"Weight Loss": random.uniform(0.0, 10.0) # Weight loss in kg

}

self.data.append(symptoms)

return self.data

def display\_sorted\_data(self, sort\_by):

df = pd.DataFrame(self.data)

sorted\_df = df.sort\_values(by=sort\_by)

print(sorted\_df)

# Example usage

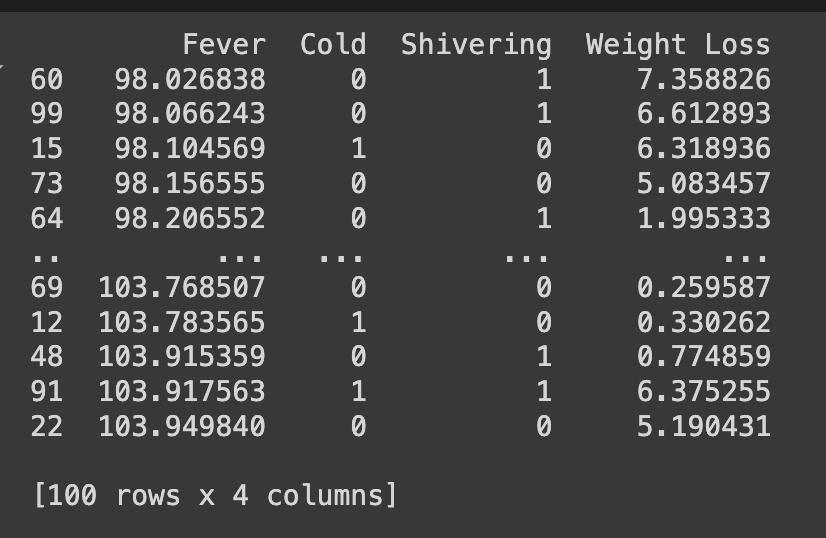
covid\_model = CovidSymptoms()

covid\_model.generate\_data()

# Sort and display by a symptom, e.g., "Fever"

covid\_model.display\_sorted\_data(sort\_by="Fever")

**Output:**



**Q:6 Generate a model to represent a mathematical equation, write a program to parse the**

**equation, and ask for input for each parameter.**

Solution: Mathematical Equation Parser

Here’s a Python model to parse a mathematical equation and request input for each variable.

Code:

class Equation:

def \_init\_(self, equation\_str):

self.equation\_str = equation\_str

self.variables = self.extract\_variables()

def extract\_variables(self):

variables = set()

for char in self.equation\_str:

if char.isalpha():

variables.add(char)

return variables

def request\_input(self):

inputs = {}

for var in self.variables:

inputs[var] = float(input(f"Enter the value for {var}: "))

return inputs

def evaluate(self, inputs):

eval\_str = self.equation\_str

for var, value in inputs.items():

eval\_str = eval\_str.replace(var, str(value))

return eval(eval\_str)

def display\_result(self):

inputs = self.request\_input()

result = self.evaluate(inputs)

print(f"The result of the equation {self.equation\_str} is: {result}")

# Example usage

equation = Equation("a \* b + c - d")

equation.display\_result()

**Output:**

