PRN = 21070521108 Name = Vedant Bhoyar

Q 1. Generate a model in Python for representation of a bank account of type savings and balance along with transactions of deposit and withdrawals and currently create a program to generate 100 accounts with Random balance and transactions for no. of months and no. of transactions with a seed value of amount. Print all 100 accounts with the last balance and organize them by lowest to highest balance.

Code:

import random

class SavingsAccount:

    def \_\_init\_\_(self, account\_id, initial\_balance):

        self.account\_id = account\_id

        self.balance = initial\_balance

        self.transactions = []

    def deposit(self, amount):

        self.balance += amount

        self.transactions.append(f"Deposit: {amount}")

    def withdraw(self, amount):

        if amount <= self.balance:

            self.balance -= amount

            self.transactions.append(f"Withdrawal: {amount}")

        else:

            self.transactions.append(f"Failed Withdrawal: {amount} (Insufficient Balance)")

    def \_\_repr\_\_(self):

        return f"Account ID: {self.account\_id}, Balance: {self.balance}, Transactions: {len(self.transactions)}"

# Function to generate accounts with random balances and transactions

def generate\_accounts(num\_accounts, months, seed\_amount):

    random.seed(seed\_amount)

    accounts = []

    for i in range(1, num\_accounts + 1):

        initial\_balance = random.randint(100, 5000)  # Random initial balance between 100 and 5000

        account = SavingsAccount(account\_id=f"AC{i:03}", initial\_balance=initial\_balance)

        # Simulate transactions for the number of months

        for \_ in range(months):

            num\_transactions = random.randint(1, 10)  # Random number of transactions per month

            for \_ in range(num\_transactions):

                if random.choice([True, False]):

                    # Random deposit

                    deposit\_amount = random.randint(50, 500)

                    account.deposit(deposit\_amount)

                else:

                    # Random withdrawal

                    withdrawal\_amount = random.randint(50, 500)

                    account.withdraw(withdrawal\_amount)

        accounts.append(account)

    return accounts

# Function to print accounts sorted by balance

def print\_sorted\_accounts(accounts):

    sorted\_accounts = sorted(accounts, key=lambda x: x.balance)

    for account in sorted\_accounts:

        print(account)

# Generate 100 accounts with random balance and transactions over 12 months

accounts = generate\_accounts(num\_accounts=100, months=12, seed\_amount=42)

# Print the accounts sorted by balance (lowest to highest)

print\_sorted\_accounts(accounts)

Output:

Account ID: AC008, Balance: 3, Transactions: 67

Account ID: AC094, Balance: 85, Transactions: 65

Account ID: AC044, Balance: 137, Transactions: 46

Account ID: AC095, Balance: 217, Transactions: 69

Account ID: AC027, Balance: 269, Transactions: 71

Account ID: AC013, Balance: 358, Transactions: 80

Account ID: AC039, Balance: 430, Transactions: 65

Account ID: AC090, Balance: 628, Transactions: 69

Account ID: AC028, Balance: 672, Transactions: 59

Account ID: AC081, Balance: 692, Transactions: 86

Account ID: AC046, Balance: 712, Transactions: 78

Account ID: AC052, Balance: 727, Transactions: 71

Account ID: AC100, Balance: 742, Transactions: 65

Account ID: AC058, Balance: 757, Transactions: 52

Account ID: AC019, Balance: 773, Transactions: 77

Account ID: AC037, Balance: 808, Transactions: 54

Account ID: AC054, Balance: 874, Transactions: 76

Account ID: AC057, Balance: 875, Transactions: 54

Account ID: AC038, Balance: 899, Transactions: 57

Account ID: AC085, Balance: 1091, Transactions: 56

Account ID: AC064, Balance: 1125, Transactions: 55

Account ID: AC071, Balance: 1141, Transactions: 75

Account ID: AC005, Balance: 1162, Transactions: 72

Account ID: AC086, Balance: 1335, Transactions: 68

Account ID: AC099, Balance: 1346, Transactions: 69

Account ID: AC076, Balance: 1460, Transactions: 64

Account ID: AC059, Balance: 1556, Transactions: 67

Account ID: AC067, Balance: 1632, Transactions: 59

Account ID: AC025, Balance: 1664, Transactions: 73

Account ID: AC015, Balance: 1665, Transactions: 55

Account ID: AC055, Balance: 1711, Transactions: 70

Account ID: AC075, Balance: 1803, Transactions: 88

Account ID: AC083, Balance: 1877, Transactions: 63

Account ID: AC004, Balance: 1911, Transactions: 82

Account ID: AC056, Balance: 1926, Transactions: 42

Account ID: AC091, Balance: 1955, Transactions: 74

Account ID: AC011, Balance: 1992, Transactions: 81

Account ID: AC089, Balance: 2103, Transactions: 72

Account ID: AC045, Balance: 2114, Transactions: 75

Account ID: AC080, Balance: 2238, Transactions: 66

Account ID: AC060, Balance: 2240, Transactions: 66

Account ID: AC051, Balance: 2256, Transactions: 60

Account ID: AC074, Balance: 2268, Transactions: 70

Account ID: AC010, Balance: 2364, Transactions: 55

Account ID: AC029, Balance: 2513, Transactions: 63

Account ID: AC006, Balance: 2601, Transactions: 81

Account ID: AC062, Balance: 2675, Transactions: 57

Account ID: AC078, Balance: 2707, Transactions: 53

Account ID: AC014, Balance: 2744, Transactions: 90

Account ID: AC066, Balance: 2751, Transactions: 63

Account ID: AC073, Balance: 2788, Transactions: 65

Account ID: AC040, Balance: 2795, Transactions: 59

Account ID: AC024, Balance: 3002, Transactions: 63

Account ID: AC016, Balance: 3039, Transactions: 68

Account ID: AC018, Balance: 3107, Transactions: 63

Account ID: AC093, Balance: 3204, Transactions: 65

Account ID: AC033, Balance: 3220, Transactions: 72

Account ID: AC077, Balance: 3279, Transactions: 55

Account ID: AC001, Balance: 3296, Transactions: 61

Account ID: AC042, Balance: 3296, Transactions: 60

Account ID: AC098, Balance: 3298, Transactions: 73

Account ID: AC061, Balance: 3330, Transactions: 53

Account ID: AC031, Balance: 3451, Transactions: 66

Account ID: AC047, Balance: 3475, Transactions: 80

Account ID: AC032, Balance: 3505, Transactions: 80

Account ID: AC021, Balance: 3552, Transactions: 60

Account ID: AC007, Balance: 3604, Transactions: 71

Account ID: AC009, Balance: 3622, Transactions: 76

Account ID: AC048, Balance: 3667, Transactions: 60

Account ID: AC092, Balance: 3702, Transactions: 50

Account ID: AC084, Balance: 3841, Transactions: 72

Account ID: AC068, Balance: 3949, Transactions: 72

Account ID: AC050, Balance: 4148, Transactions: 58

Account ID: AC072, Balance: 4170, Transactions: 65

Account ID: AC022, Balance: 4266, Transactions: 65

Account ID: AC070, Balance: 4432, Transactions: 77

Account ID: AC063, Balance: 4442, Transactions: 56

Account ID: AC097, Balance: 4616, Transactions: 68

Account ID: AC030, Balance: 4693, Transactions: 63

Account ID: AC087, Balance: 4842, Transactions: 61

Account ID: AC036, Balance: 4904, Transactions: 69

Account ID: AC026, Balance: 5080, Transactions: 36

Account ID: AC065, Balance: 5132, Transactions: 72

Account ID: AC088, Balance: 5179, Transactions: 56

Account ID: AC035, Balance: 5334, Transactions: 70

Account ID: AC002, Balance: 5488, Transactions: 76

Account ID: AC079, Balance: 5857, Transactions: 74

Account ID: AC023, Balance: 6044, Transactions: 59

Account ID: AC020, Balance: 6151, Transactions: 71

Account ID: AC082, Balance: 6277, Transactions: 49

Account ID: AC041, Balance: 6307, Transactions: 76

Account ID: AC012, Balance: 6328, Transactions: 70

Account ID: AC043, Balance: 6330, Transactions: 57

Account ID: AC003, Balance: 6407, Transactions: 55

Account ID: AC034, Balance: 6490, Transactions: 74

Account ID: AC017, Balance: 6541, Transactions: 60

Account ID: AC069, Balance: 7412, Transactions: 72

Account ID: AC096, Balance: 8239, Transactions: 65

Account ID: AC053, Balance: 8859, Transactions: 54

Account ID: AC049, Balance: 8933, Transactions: 67

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.

import pandas as pd

import random

# Function to generate random values for symptoms

def generate\_data(num\_samples=100):

    data = {

        'fever': [round(random.uniform(98.0, 104.0), 1) for \_ in range(num\_samples)],  # Temperature in Fahrenheit

        'cold': [random.randint(0, 1) for \_ in range(num\_samples)],  # 0 = No cold, 1 = Cold

        'shivering': [random.randint(0, 1) for \_ in range(num\_samples)],  # 0 = No shivering, 1 = Shivering

        'weight\_loss': [round(random.uniform(0, 10), 1) for \_ in range(num\_samples)],  # Weight loss in pounds

    }

    return pd.DataFrame(data)

# Function to sort data based on input parameter

def sort\_by\_parameter(df, parameter):

    if parameter in df.columns:

        return df.sort\_values(by=[parameter])

    else:

        print(f"Invalid parameter: {parameter}")

        return df

# Generate 100 random model data

df = generate\_data()

# Input parameter for sorting

parameter = input("Enter the parameter to sort by (fever, cold, shivering, weight\_loss): ")

# Sort and display the data

sorted\_df = sort\_by\_parameter(df, parameter)

print(sorted\_df)

**Output:**

Enter the parameter to sort by (fever, cold, shivering, weight\_loss): fever

fever cold shivering weight\_loss

3 98.0 0 1 6.7

50 98.1 0 1 7.2

45 98.1 0 1 5.1

77 98.2 1 0 7.0

79 98.6 0 1 9.9

.. ... ... ... ...

80 103.9 0 1 2.2

44 103.9 0 1 7.8

32 103.9 0 1 9.0

65 104.0 1 1 0.6

57 104.0 0 1 0.5

[100 rows x 4 columns]

Enter the parameter to sort by (fever, cold, shivering, weight\_loss): shivering

fever cold shivering weight\_loss

63 102.6 0 0 3.8

25 103.7 0 0 9.2

85 102.3 1 0 3.6

84 103.7 1 0 3.6

28 103.9 1 0 0.6

.. ... ... ... ...

39 99.9 0 1 0.3

38 103.8 1 1 3.0

37 102.0 0 1 2.0

47 100.5 0 1 1.6

99 98.4 1 1 8.7

[100 rows x 4 columns]

Enter the parameter to sort by (fever, cold, shivering, weight\_loss): weight\_loss

fever cold shivering weight\_loss

78 103.9 1 0 0.1

84 103.9 0 0 0.1

22 99.2 0 0 0.1

44 100.4 1 0 0.2

27 98.5 1 0 0.4

.. ... ... ... ...

87 101.7 1 1 9.2

0 99.4 0 0 9.4

39 103.0 1 1 9.5

55 103.6 0 1 9.5

66 99.4 0 0 10.0

Enter the parameter to sort by (fever, cold, shivering, weight\_loss): cold

fever cold shivering weight\_loss

0 101.1 0 0 7.8

66 99.6 0 0 1.4

64 101.8 0 0 5.9

63 101.1 0 0 4.5

62 100.1 0 0 7.5

.. ... ... ... ...

21 103.2 1 0 4.6

20 101.6 1 1 7.8

68 99.7 1 1 4.0

73 101.4 1 1 4.0

99 103.0 1 0 0.1