FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERIG.
Fr. Agnel Ashram, Bandstand, Bandra (W) Mumbai 400050

PROGRAMMING ASSIGNMENT 2

Course: Skill Based Programming – Python programming

Class: S.E. Computer (Division A and B)
Date of Assignment: 08-04-2024
Date of Submission: 15-04-2024

Question	CO Mapping
Q1	CSL405.5 Develop program using multithreading and networking concepts.
Q2	CSL405.6 Develop real world application using frameworks and in built libraries in python.

Q1) You are tasked with designing a program to simulate a banking system that manages customer accounts. The program should utilize multiple threads to perform a resource-intensive task of processing transactions concurrently. However, it is crucial to ensure that the threads are synchronized properly to avoid race conditions and maintain data integrity.

Requirements:

1. Implement a class Bank that represents the banking system. This class should have methods to perform deposit, withdrawal, and balance inquiry operations on customer accounts.

2. Each customer account should be represented by a unique identifier (e.g., account number) and should maintain its balance.

3. The program should support concurrent processing of transactions by multiple threads.

4. Implement methods to perform deposit and withdrawal transactions, ensuring proper synchronization to prevent race conditions.

5. Implement a method to check the balance of a customer account.

6. Ensure that the program handles concurrent transactions safely, maintaining data integrity and consistency.

7. Test the program with multiple threads performing simultaneous transactions, and verify that the results are correct and consistent.

```python
from threading import Thread, Lock




class Bank:
```

```python
    def __init__(self):

        self.accounts = {}

        self.lock = Lock()


    def checkBalance(self, accountNumber):

        with self.lock:

            return self.accounts.get(accountNumber, 0)


    def depositMoney(self, accountNumber, amount):

        with self.lock:

            current_balance = self.accounts.get(accountNumber, 0)

            self.accounts[accountNumber] = current_balance + amount

            print(f"Deposited {amount} into account {accountNumber}")


    def withdrawMoney(self, accountNumber, amount):

        with self.lock:

            current_balance = self.accounts.get(accountNumber, 0)

            if current_balance >= amount:

                self.accounts[accountNumber] = current_balance - amount

                print(f"{amount} withdrawn successfully. Current balance is
{self.accounts.get(accountNumber)}")

            else:

                print("Not enough balance. Current balance is ",
current_balance)
```

```python
    def simulateTransaction(self, accountNumber,
numberOfTransactionsPerThread):

        deposit_amount = 1000

        withdrawal_amount = 200


        for i in range(numberOfTransactionsPerThread):

            self.depositMoney(accountNumber, deposit_amount)

            self.withdrawMoney(accountNumber, withdrawal_amount)


if __name__ == "__main__":

    bank = Bank()

    accountNumber = 395795739474

    numberOfThreads = 5

    numberOfTransactionsPerThread = 2

    threads = []


    for _ in range(numberOfThreads):

        t = Thread(target=bank.simulateTransaction, args=(accountNumber,
numberOfTransactionsPerThread))

        threads.append(t)

        t.start()


    for t in threads:
```

```
        t.join()


   print("Total current balance: ", bank.checkBalance(accountNumber))
```

```
SyntaxError: leading zeros in decimal integer literals are r
PS C:\Users\Mark Lopes\Desktop\college\Sem_4\Python> python
Deposited 1000 into account 395795739474
200 withdrawn successfully. Current balance is 800
Deposited 1000 into account 395795739474
200 withdrawn successfully. Current balance is 1600
Deposited 1000 into account 395795739474
200 withdrawn successfully. Current balance is 2400
Deposited 1000 into account 395795739474
200 withdrawn successfully. Current balance is 3200
Deposited 1000 into account 395795739474
200 withdrawn successfully. Current balance is 4000
Deposited 1000 into account 395795739474
200 withdrawn successfully. Current balance is 4800
Deposited 1000 into account 395795739474
200 withdrawn successfully. Current balance is 5600
Deposited 1000 into account 395795739474
200 withdrawn successfully. Current balance is 6400
Deposited 1000 into account 395795739474
200 withdrawn successfully. Current balance is 7200
Deposited 1000 into account 395795739474
200 withdrawn successfully. Current balance is 8000
Total current balance:  8000
PS C:\Users\Mark Lopes\Desktop\college\Sem_4\Python>
```

Q2) Create a Grade book of the student using Pandas
   Objectives: To learn
   · Load and merge data from multiple sources with pandas
   · Filter and group data in a pandas DataFrame
   · Calculate and plot grades in a pandas DataFrame

Problem statement:
There are three categories of assignments that you had in your class:

Second Year Computer Engineering (Sem – IV) Python Programming (CSL405)

FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERIG.

1. Exams
2. Homework
3. Quizzes
Create three different files (.csv) for grades

Quiz.csv

Roll No	Name	Quiz1	Max Score	Quiz2	Max Score
101	Ajay	8	10	12	15
102	Rita	9	10	13	15
102	Shalmali	10	10	14	15

Homework.csv

Roll No	Name	HW1	Max Score	HW2	Max Score
101	Ajay	NaN	15	9	10
102	Rita	12	15	Nan	10
102	Shalmali	12	15	6	10

Exam.csv (For simplicity assume all tests are of 25M)

Roll No	Name	Test1	Test2
101	Ajay	23	Nan
102	Rita	22	22
102	Shalmali	21	21

You need to create a Final Grading sheet

Roll No	Name	Quizzes	Homework	Exam	Final Score	Final Grade
101	Ajay	20	09	23		
102	Rita	22	12	44		
102	Shalmali	24	18	42		

Steps Involved:
1) Read all three sources of data - quiz marks, Homework marks and Exam marks. 2) Merge the data frames. Now that you have all your data loaded, you can combine the data from your three Data Frames. You'll merge the data together in two steps:
  Merge Quiz and homework gardes together into a new DataFrame called final_data.
  Merge final_data and Exam_grades together.
3) Fill the Nan values if any. (In this case Nan values will be filled with Zero)
4) Calculate Grades with pandas

Second Year Computer Engineering (Sem – IV) Python Programming (CSL405)

Calculate Quiz Score

Roll No	Name	Sum of Quiz scores	Sum of Max Score	Total Quiz score
101	Ajay	20	25	0.80
102	Rita	22	25	0.88
102	Shalmali	24	25	0.96

Calculate Homework score:

Roll No	Name	Sum of Homework scores	Sum of Max scores	Total Homework score
101	Ajay	09	25	0.36
102	Rita	12	25	0.48
102	Shalmali	18	25	0.72

Calculate Exam score:

Roll No	Name	Sum of Exam score	Sum of Max Score	Total Exam score
101	Ajay	23	50	0.46
102	Rita	44	50	0.88
102	Shalmali	42	50	0.84

Each of these categories is assigned a <u>weight</u> toward the students' final score. Use following weight assignment.

Category	Weight	
Quiz	30%	0.3
Homework	30%	0.3

Exam	40%	0.4

The final score can be calculated by multiplying the weight by the total score from each category and summing all these values. The final score will then be converted to 100 and grades can be assigned according to following rules

Roll No	Name	Quiz	HW	Exam	Final Grade
101	Ajay	0.80*0.3= 0.24	0.36*0.3= 0.108	0.46*0.4= 0.184	0.53
102	Rita	0.88*0.3=0.264	0.48*0.3=0.144	0.88*0.4= 0.352	0.76
102	Shalmali	0.96*0.3=0.288	0.72*0.3=0.216	0.84*0.4=0.33	0.83

5) Finally Plot the grades using pandas plot (Optional)

References:

1) Connecting Django with Mysql:://.youtube.com/watch?v=8gSjvehTqAk
2) Creating Grade book:

https://realpython.com/pandas-project-gradebook/#:~:text=the%20link%20below%3A-,Get%20the%20Source%20Code%3A%20Click%20here%20to%20get%20the%20source,files%20for%20your%20gradebook%20script.

Q2

April 15, 2024

```python
import pandas as pd
import numpy as np
Quiz_data = pd.read_csv('Quiz.csv')
Homework_data = pd.read_csv('Homework.csv')
Exam_data = pd.read_csv('Exam.csv')
```

```python
final_data = pd.merge(Quiz_data, Homework_data, on=['Roll␣
 ↪No','Name'],how="outer")
final_data=pd.merge(final_data, Exam_data,on=["Roll No","Name"],how="outer")
final_data
```

```
   Roll No     Name  Quiz1  Max Score_x  Quiz2  Max Score.1_x   HW1  \
0      101     Ajay      8           10     12             15   NaN
1      102     Rita      9           10     13             15  12.0
2      102  Shalmali     10          10     14             15  12.0

   Max Score_y  HW2  Max Score.1_y  Test1  Test2
0           15    9             10     23    Nan
1           15  Nan             10     22     22
2           15    6             10     21     21
```

```python
final_data.fillna(0, inplace = True)
final_data
```

```
   Roll No     Name  Quiz1  Max Score_x  Quiz2  Max Score.1_x   HW1  \
0      101     Ajay      8           10     12             15   0.0
1      102     Rita      9           10     13             15  12.0
2      102  Shalmali     10          10     14             15  12.0

   Max Score_y  HW2  Max Score.1_y  Test1  Test2
0           15    9             10     23    Nan
1           15  Nan             10     22     22
2           15    6             10     21     21
```

```python
final_data["Sum of Quiz Scores"] = final_data["Quiz1"] + final_data["Quiz2"]
final_data["Sum of Max Scores"] = final_data["Max Score_x"] + final_data["Max␣
 ↪Score.1_x"]
```

```python
new_quiz = final_data[['Roll No', 'Name', 'Sum of Quiz Scores', 'Sum of Max␣
  ↪Scores']]

new_quiz['Total Quiz Score'] = new_quiz['Sum of Quiz Scores'] / new_quiz['Sum␣
  ↪of Max Scores']

new_quiz
```

C:\Users\Mark Lopes\AppData\Local\Temp\ipykernel_16992\3359108876.py:6:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_quiz['Total Quiz Score'] = new_quiz['Sum of Quiz Scores'] / new_quiz['Sum
of Max Scores']

```
[ ]:    Roll No      Name  Sum of Quiz Scores  Sum of Max Scores  Total Quiz Score
    0       101      Ajay                  20                 25              0.80
    1       102      Rita                  22                 25              0.88
    2       102  Shalmali                  24                 25              0.96
```

```python
[ ]: # Calculate final homework and final max homework scores
    final_data["HW1"] = pd.to_numeric(final_data["HW1"], errors='coerce') # coz␣
      ↪directly its showing error
    final_data["HW2"] = pd.to_numeric(final_data["HW2"], errors='coerce')
    final_data['HW2'].fillna(0, inplace=True)#coz hw2 still hd Nan

    # Calculate final homework and final max homework scores
    final_data["Sum of Hw Scores"] = final_data["HW1"] + final_data["HW2"]
    final_data["Sum of Maximum Scores"] = final_data["Max Score_y"] +␣
      ↪final_data["Max Score.1_y"]
    new_homework= final_data[['Roll No', 'Name', 'Sum of Hw Scores','Sum of Maximum␣
      ↪Scores']]
    new_homework['Total Homework Score'] = (new_homework['Sum of Hw Scores'] /␣
      ↪new_homework['Sum of Maximum Scores'])
    new_homework
```

C:\Users\Mark Lopes\AppData\Local\Temp\ipykernel_16992\3477659495.py:4:
FutureWarning: A value is trying to be set on a copy of a DataFrame or Series
through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)

instead, to perform the operation inplace on the original object.

```
final_data['HW2'].fillna(0, inplace=True)#coz hw2 still hd Nan
C:\Users\Mark Lopes\AppData\Local\Temp\ipykernel_16992\3477659495.py:10:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_homework['Total Homework Score'] = (new_homework['Sum of Hw Scores'] /
new_homework['Sum of Maximum Scores'])
```

```
[ ]:    Roll No      Name  Sum of Hw Scores  Sum of Maximum Scores  \
     0      101      Ajay               9.0                     25
     1      102      Rita              12.0                     25
     2      102   Shalmali             18.0                     25

        Total Homework Score
     0                  0.36
     1                  0.48
     2                  0.72
```

```python
[ ]: final_data["Test1"] = pd.to_numeric(final_data["Test1"], errors='coerce') # coz␣
       ↪directly its showing error
     final_data["Test2"] = pd.to_numeric(final_data["Test2"], errors='coerce')
     final_data['Test2'].fillna(0, inplace=True)#coz hw2 still hd Nan

     # Calculate final homework and final max homework scores
     final_data["Sum of Exam Score"] = final_data["Test1"] + final_data["Test2"]
     final_data['Sum of Max Score'] = [50,50,50]
     new_exam= final_data[['Roll No', 'Name', 'Sum of Exam Score','Sum of Max␣
       ↪Score']]
     new_exam['Total Exam Score'] = (new_exam['Sum of Exam Score'] / new_exam['Sum␣
       ↪of Max Score'])
     new_exam
```

```
C:\Users\Mark Lopes\AppData\Local\Temp\ipykernel_16992\505153484.py:3:
FutureWarning: A value is trying to be set on a copy of a DataFrame or Series
through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.
```

```
final_data['Test2'].fillna(0, inplace=True)#coz hw2 still hd Nan
```
C:\Users\Mark Lopes\AppData\Local\Temp\ipykernel_16992\505153484.py:9:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_exam['Total Exam Score'] = (new_exam['Sum of Exam Score'] / new_exam['Sum
of Max Score'])

```
[ ]:    Roll No      Name  Sum of Exam Score  Sum of Max Score  Total Exam Score
     0      101      Ajay               23.0                50              0.46
     1      102      Rita               44.0                50              0.88
     2      102  Shalmali               42.0                50              0.84
```

```python
[ ]: # Assign weights to each category
     weights = {'Total Quiz Score': 0.3, 'Total Homework Score': 0.3, 'Total Exam␣
      ↪Score': 0.4}

     # Calculate the weighted scores for each category
     new_quiz['Weighted_Quiz'] = new_quiz['Total Quiz Score'] * weights['Total Quiz␣
      ↪Score']
     new_homework['Weighted_HW'] = new_homework['Total Homework Score'] *␣
      ↪weights['Total Homework Score']
     new_exam['Weighted_Exam'] = new_exam['Total Exam Score'] * weights['Total Exam␣
      ↪Score']

     # Concatenate the DataFrames along the columns axis (horizontally)
     final = pd.concat([new_quiz['Weighted_Quiz'], new_homework['Weighted_HW'],␣
      ↪new_exam['Weighted_Exam']], axis=1)

     # Calculate the final score by summing up the weighted scores for all categories
     final['Final_Score'] = final.sum(axis=1)

     # Display the final DataFrame
     final
```

C:\Users\Mark Lopes\AppData\Local\Temp\ipykernel_16992\2619076866.py:5:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_quiz['Weighted_Quiz'] = new_quiz['Total Quiz Score'] * weights['Total Quiz

```
Score']
C:\Users\Mark Lopes\AppData\Local\Temp\ipykernel_16992\2619076866.py:6:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_homework['Weighted_HW'] = new_homework['Total Homework Score'] *
weights['Total Homework Score']
C:\Users\Mark Lopes\AppData\Local\Temp\ipykernel_16992\2619076866.py:7:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_exam['Weighted_Exam'] = new_exam['Total Exam Score'] * weights['Total Exam
Score']
```

```
[ ]:    Weighted_Quiz  Weighted_HW  Weighted_Exam  Final_Score
    0           0.240        0.108          0.184        0.532
    1           0.264        0.144          0.352        0.760
    2           0.288        0.216          0.336        0.840
```