

## 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.

```
from threading import Thread, Lock
```

```
class Bank:
```

```
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            print(f"{amount} withdrawn successfully.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)
```

```
    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 ENGINEERING.

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 grades 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)

FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING.

Fr. Agnel Ashram, Bandstand, Bandra (W) Mumbai 400050

#### 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 weight 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](https://www.youtube.com/watch?v=8gSjvehTqAk):[://.youtube.com/watch?v=8gSjvehTqAk](https://www.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.>