

Please use Python 3.9 to write a code.
This assignment is due on February 4th at 11:59 PM.
Thank you very much for your help.

Assignment Specifications:

Download the file **bank_account.py** first (the code is given below, and it just has 8 lines).

bank_account.py:

```
class BankAccount:
    def __init__(self, name):
        self.name = name
        self.balance = 0.0

    def deposit(self, amount):
        self.balance += amount

    def withdraw(self, amount):
        self.balance -= amount
```

BankAccount:

Using the code in the file above as a starting point, add the functionality described below.

1. Modify the code to be encapsulated with private data attributes and read-only properties representing the **name** and **balance**.
2. Modify the **BankAccount** class to enforce the requirement that the account's balance can never become negative. This means you should forbid negative deposits and any withdrawals that exceed the account's balance.
3. Add a property to the **BankAccount** class named **transaction_fee** for a real number representing an amount of money to be deducted as a fee to the bank every time the user withdraws money. The default value is \$0.00, but the client can change the value. Deduct the transaction fee during every call to **withdraw()** but not from deposits. Make sure the balance cannot go negative during a withdrawal. If the withdrawal (amount plus transaction fee) would cause the account balance to become negative, do not modify the account balance. Instead, raise a *custom error type* (see here and here for more information). Your custom error should be defined as an *inner class* to **BankAccount**.

4. Add a **__str__** method to the **BankAccount** class. This method should return a string that contains the account's name and balance separated by a comma and space. For example, if an account object named meriweather has the name **Meriweather** and a balance of \$3.70, the call **str(meriweather)** should return the string "Meriweather, \$3.70".
5. Add a **transfer** method to the **BankAccount** class. The method should move money from the current bank account to another account. The method accepts two parameters aside from **self**: another instance of **BankAccount** and a real number for the amount of money to transfer. There is a \$5 fee for transferring money that is to be deducted from the current account's balance before any transfer happens. The method should modify the two **BankAccount** objects such that the current object has its balance decreased by the given amount plus the \$5 fee and the other account's balance is increased by the given amount. If the current account object does not have enough money to make the full transfer, transfer whatever money is left in the account after the \$5 is deducted. If the account has under \$5 or the amount to transfer is \$0 or less, no transfer should occur and neither account's state should be modified.
6. Add the necessary docstring and encoding line to the beginning of the module, and add docstrings at the module, class, and method levels.
7. Generate a separate file named bank_account_driver.py that imports your custom type, generates a few instances of it, and exercises its functionality printing messages as necessary to show its workings. This file should also have the standard required lines of code at its beginning.