Class Exercise:

Exception v2

You are tasked with creating a class to represent a bank account. The class should have the following functionalities:

1. The account should keep track of the account holder's name and the current balance.
2. It should have a method called "deposit" that allows depositing money into the account.
3. The deposit amount should be a positive number. If a negative or zero amount is provided, it should raise a custom exception called **"InvalidDepositAmountError"** with the message "Invalid deposit amount: <amount>".
4. It should have a method called "withdraw" that allows withdrawing money from the account.
5. The withdrawal amount should be a positive number and should not exceed the current balance. If a negative amount is provided or the withdrawal amount exceeds the balance, it should raise a custom exception called **"InsufficientFundsError"** with the message "Insufficient funds for withdrawal: <amount>".
6. The class should implement exception handling and a finally block to ensure that the account balance is always updated correctly, even if an exception is raised.

class InvalidDepositAmountError(Exception):

    pass

class InsufficientFundsError(Exception):

    pass

class BankAccount:

    def \_\_init\_\_(self, account\_holder, initial\_balance=0):

# Your code here

    def deposit(self, amount):

# Your code here

    def withdraw(self, amount):

# Your code here

    def get\_balance(self):

# Your code here

# Test the BankAccount class

# Your code here

Expected output:

Initial balance: 1000

Insufficient funds for withdrawal: 2000

Invalid withdrawal amount: -100

Current balance: 1500