```
'''Implement a class called BankAccount that represents a bank account. The class should have private
attributes for account number, account holder name, and account balance. Include methods to
deposit money, withdraw money, and display the account balance. Ensure that the account balance
cannot be accessed directly from outside the class. Write a program to create an instance of the
BankAccount class and test the deposit and withdrawal fanctionality.'''


class BankAccount:

 def __init__(self, account_number, account_holder_name, initial_balance=0.0):
   self.__account_number = account_number
   self.__account_holder_name = account_holder_name
   self.__account_balance = initial_balance

 def deposit(self, amount):
   if amount > 0:
     self.__account_balance += amount
     #self.__account_balance = self.__account_balance+amount
     print("Deposited ₹{}. New balance: ₹{}".format(amount,
                                        self.__account_balance))
   else:
       print("Invalid deposit amount.")

 def withdraw(self, amount):
   if amount > 0 and amount <= self.__account_balance:
     self.__account_balance -= amount
     #self.__account_balance = self.__account_balance_amount
     print("Withdrew ₹{}. New balance: ₹{}".format(amount,
                                       self.__account_balance))
   else:
     print("Invalid withdrawal amount or insufficient balance.")

 def display_balance(self):
   print("Account balance for {} (Account #{}): ₹{}".format(
       self.__account_holder_name, self.__account_number,
       self.__account_balance))


#create an instance of the BankAccount class
account = BankAccount(account_number="123456789",
                      account_holder_name="Shakira",
                      initial_balance=5000.0)

#Test deposit and withdraw functionality
account.display_balance()
account.deposit(500.0)
account.withdraw(200.0)
account.withdraw(20000.0)
account.display_balance()
```