'''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 functionality.'''

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="Hari Prabu",

initial\_balance=5000.0)

# Test deposit and withdrawal functionality

account.display\_balance()

account.deposit(500.0)

account.withdraw(200.0)

account.withdraw(20000.0)

account.display\_balance()