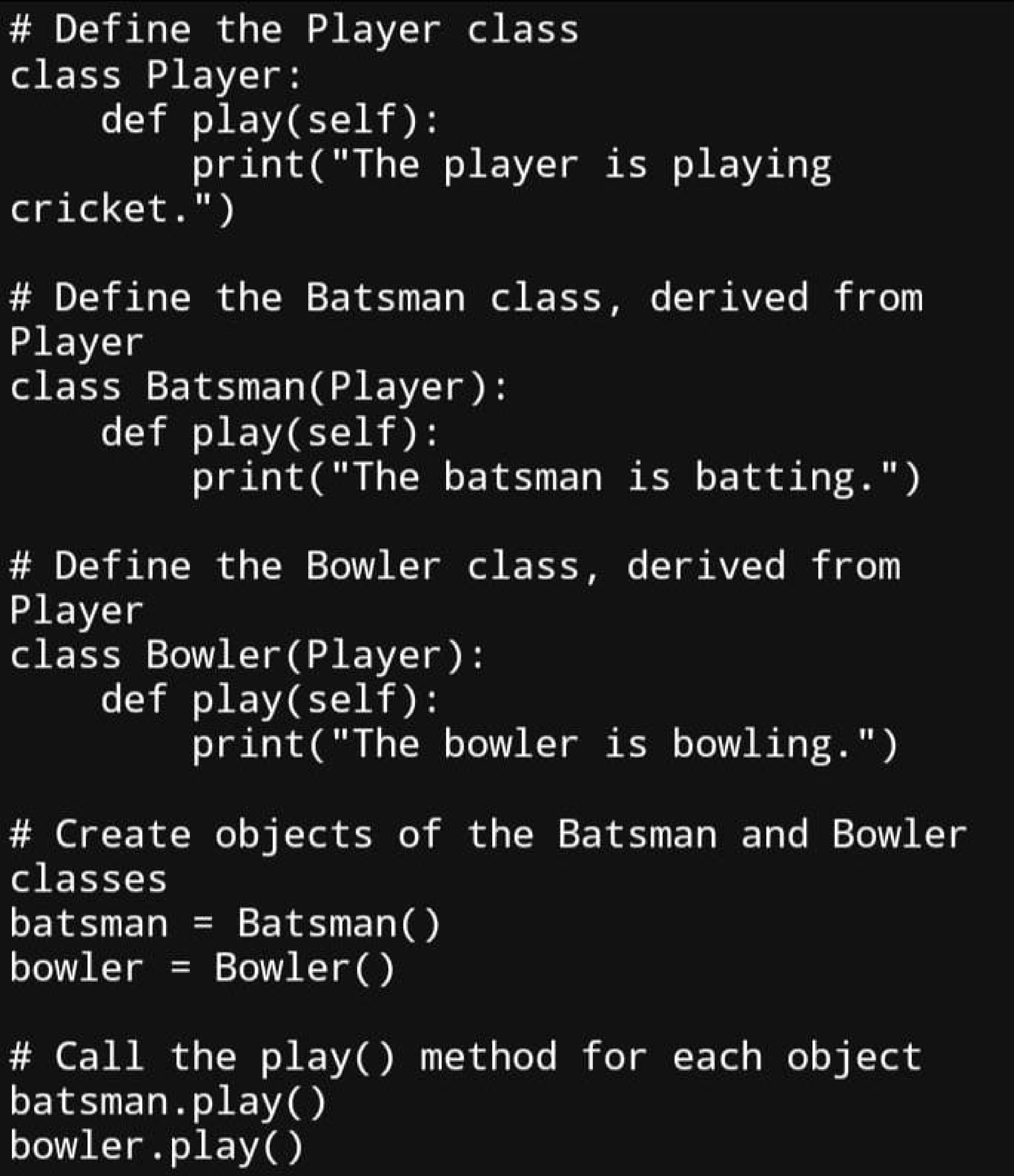


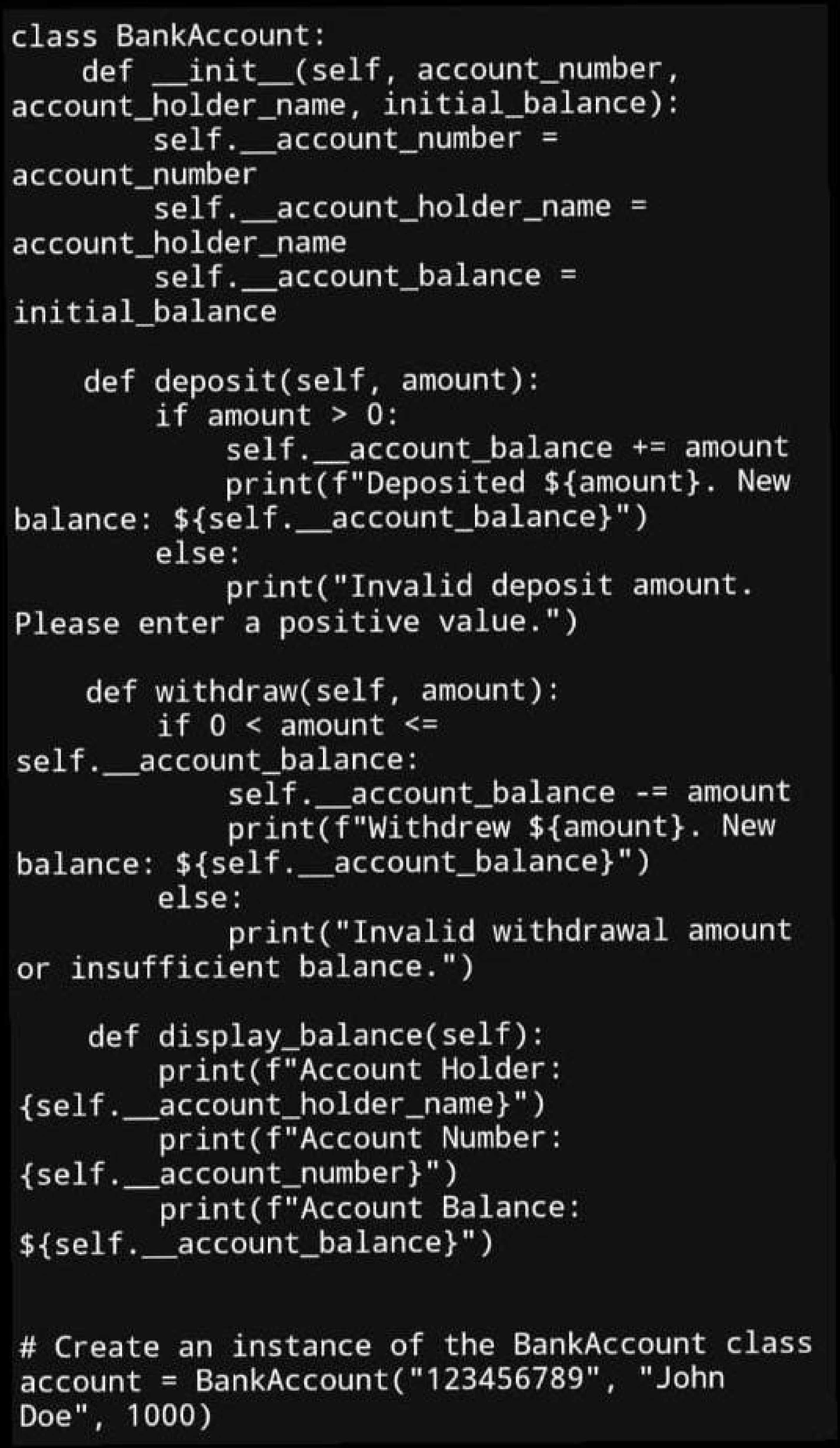
```
# Define the Player class
class Player:
    def play(self):
        print("The player is playing
cricket.")

# Define the Batsman class, derived from
Player
class Batsman(Player):
    def play(self):
        print("The batsman is batting.")

# Define the Bowler class, derived from
Player
class Bowler(Player):
    def play(self):
        print("The bowler is bowling.")

# Create objects of the Batsman and Bowler
classes
batsman = Batsman()
bowler = Bowler()

# Call the play() method for each object
batsman.play()
bowler.play()
```



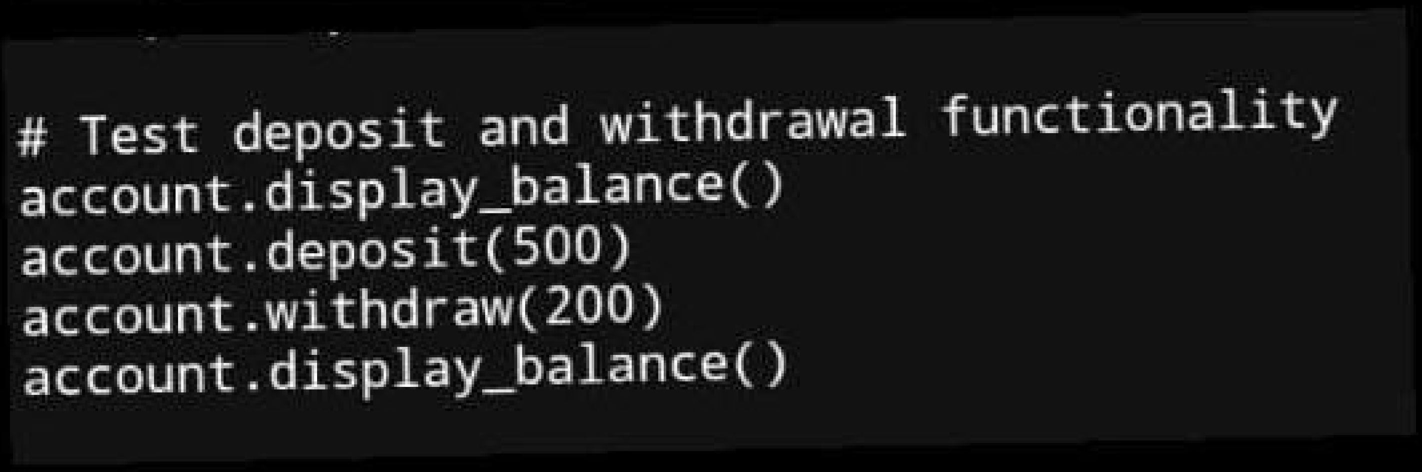
```
class BankAccount:
    def __init__(self, account_number,
account_holder_name, initial_balance):
        self.__account_number =
account_number
        self.__account_holder_name =
account_holder_name
        self.__account_balance =
initial_balance

    def deposit(self, amount):
        if amount > 0:
            self.__account_balance += amount
            print(f"Deposited ${amount}. New
balance: ${self.__account_balance}")
        else:
            print("Invalid deposit amount.
Please enter a positive value.")

    def withdraw(self, amount):
        if 0 < amount <=
self.__account_balance:
            self.__account_balance -= amount
            print(f"Withdrew ${amount}. New
balance: ${self.__account_balance}")
        else:
            print("Invalid withdrawal amount
or insufficient balance.")

    def display_balance(self):
        print(f"Account Holder:
{self.__account_holder_name}")
        print(f"Account Number:
{self.__account_number}")
        print(f"Account Balance:
${self.__account_balance}")


# Create an instance of the BankAccount class
account = BankAccount("123456789", "John
Doe", 1000)
```



```
# Test deposit and withdrawal functionality
account.display_balance()
account.deposit(500)
account.withdraw(200)
account.display_balance()
```