



Python2.1 :

Exit

```
1 class BankAccount:
2
3     def __init__(self, account_number,
4 account_holder_name, initial_balance=0.0):
5         self.__account_number=account_number
6
7         self.__account_holder_name=account_holder_name
8         self.__account_balance= initial_balance
9
10    def deposit(self, amount):
11        if amount > 0:
12            self.__account_balance += amount
13            # self.__account_balance =
14            self.__account_balance+amount
15            print("Deposited ₹{}. New balance: ₹
16 {}".format(amount,
17 self.__account_balance))
18        else:
19            print("Invalid deposit amount. please
20 deposit a positive amount.")
21
22    def withdraw (self, amount):
23        if amount > 0 and amount
24 <=self.__account_balance:
25            self.__account_balance -= amount
26            # self.__account_balance =
27            self.__account_balance - amount
28            print("withdrew ₹{}. Newbalance: ₹
29 {}".format(amount,
30 self.__account_balance))
31        else:
32            print("Invalid withdrawal amount or
33 insufficient balance.")
34
35    def display_balance(self):
36        print("Account balance for {} (Account #
37 {}): ₹{}".format(
38 self.__account_number,
39 self.__account_holder_name,
40 self.__account_balance))
```

Ln 1, Col 1 • Spaces: 2 History

main.py



Run





Python2.1 :

Exit

```
14     else:
15         print("Invalid deposit amount. please
deposit a positive amount.")
16
17     def withdraw (self, amount):
18         if amount > 0 and amount
<=self.__account_balance:
19             self.__account_balance -= amount
20             # self.__account_balance =
self.__account_balance - amount
21             print("withdrew ₹{}. Newbalance: ₹
{}".format(amount,
22                 self.__account_balance))
23         else:
24             print("Invalid withdrawal amount or
insufficient balance.")
25
26     def display_balance(self):
27         print("Account balance for {} (Account #
{}): ₹{}".format(
28             self.__account_holder_name,
self.__account_number,
29             self.__account_balance))
30
31
32 # Create an instance of the BankAccount class
33 account=BankAccount(account_number="123456789",
34     account_holder_name="sumithra",
35     initial_balance=5000.0)
36 # Test deposit and withdrawal functionality
37 account.display_balance()
38 account.deposit(500.0)
39 account.withdraw(200.0)
40 account.display_balance()
```

Ln 1, Col 1 • Spaces: 2 History

main.py



Run





Python2.1 :

Exit

Run

507ms on 20:38:13, 10/19 ✓

```
Account balance for sumithra (Account #123456789):  
₹5000.0  
Deposited ₹500.0. New balance: ₹5500.0  
withdrew ₹200.0. Newbalance: ₹5300.0  
Account balance for sumithra (Account #123456789):  
₹5300.0
```



>_ Console



Run





```
1  # define the base class Player
2  class Player:
3      def play(self):
4          print("The player is playing cricket.")
5
6  # define the derived class Batsman
7  class Batsman(Player):
8      def play(self):
9          print("The batsman is batting.")
10
11 # define the derived class Bowler
12 class Bowler(Player):
13     def play(self):
14         print("The bowler is bowling.")
15
16 # Create objects of Batsman and Bowler classes
17 batsman = Batsman()
18 bowler = Bowler()
19
20 # call the play() method for each object
21 batsman.play()
22 bowler.play()
```

Ln 1, Col 1 • Spaces: 2 History



main.py



Run

