Python2. 1 ⋮ Exit

```
1  class BankAccount:
2
3    def __init__(self,
     account_number,
     account_holder_name,
     initial_balance=0.0):
4        self.__account_number =
     account_number
5        self.__account_holder_name =
     account_holder_name
6        self.__account_balance =
     initial_balance
7
8    def deposit(self, amount):
9        if amount > 0:
10           self.__account_balance +=
     amount
11           # self.__account_balance =
     self.__account_balance+amount
12           print("Deposited ₹{}. New
     balance: ₹{}".format(amount,
13
```

Ln 1, Col 1 • Spaces: 2 History

main.py ⋮

Run

```
13

    self.__account_balance))
14 v     else:
15           print("Invalid deposit
    amount. please deposit a positive
    amount.")
16
17 v   def withdraw(self, amount):
18 v     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 v     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="12345678
   9",
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()
```

Python2. 1 Exit

Run 736ms on 23:07:37, 10/18

```
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

Python2. 2 Exit

```
# define the base class Player
class Player:

    def play(self):
        print("The player is playing cricket.")


# define the derived class Batsman
class Batsman(Player):

    def play(self):
        print("The batsman is batting.")


# define the derived class Bowler
class Bowler(Player):

    def play(self):
        print("The bowler is bowling.")


# Create objects of Batsman and
```

Ln 1, Col 1 • Spaces: 2 History

main.py

Run

Python2. 2 Exit

```
13
14
15  # define the derived class Bowler
16  class Bowler(Player):
17
18      def play(self):
19          print("The bowler is bowling.")
20
21
22  # Create objects of Batsman and
    Bowler classes
23  batsman = Batsman()
24  bowler = Bowler()
25
26  # call the play() method for each
    object
27  batsman.play()
28  bowler.play()
29
```

Ln 1, Col 1 • Spaces: 2 History

main.py

Run

Python2. 2

Exit

Run

56ms on 23:08:32, 10/18

```
The batsman is batting.
The bowler is bowling.
```

Console

Run