Challenge2.1 Exit

```
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 amount > 0 and amount <= self.__account_balance:
            self.__account_balance -= amount
            print(f"Withdrew ${amount}. New balance:
```

Ln 30, Col 57 History

main.py

Run

Challenge2.1 Exit

```
18          else:
19              print("Invalid withdrawal amount or insufficient balance.")
20
21      def display_balance(self):
22          print(f"Account Holder: {self.__account_holder_name}")
23          print(f"Account Number: {self.__account_number}")
24          print(f"Account Balance: ${self.__account_balance}")
25
26
27  # Testing the BankAccount class
28  if __name__ == "__main__":
29      # Create an instance of BankAccount
30      my_account = BankAccount("123456789", "EGAN", 100000.0)
31
32      # Display initial balance
33      my_account.display_balance()
34
35      # Deposit money
36      my_account.deposit(500.0)
37
38      # Withdraw money
39      my_account.withdraw(200.0)
40
```

Ln 30, Col 57 History

main.py

Run

Challenge2.1 Exit

```
{self.__account_holder_name}")
23          print(f"Account Number:
{self.__account_number}")
24          print(f"Account Balance:
${self.__account_balance}")
25
26
27  # Testing the BankAccount class
28  if __name__ == "__main__":
29      # Create an instance of BankAccount
30      my_account =
BankAccount("123456789", "EGAN",
100000.0)
31
32      # Display initial balance
33      my_account.display_balance()
34
35      # Deposit money
36      my_account.deposit(500.0)
37
38      # Withdraw money
39      my_account.withdraw(200.0)
40
41      # Display updated balance
42      my_account.display_balance()
```

Ln 30, Col 57 History

main.py

Run

```
Account Holder: EGAN
Account Number: 123456789
Account Balance: $100000.0
Deposited $500.0. New balance: $100500.0
Withdrew $200.0. New balance: $100300.0
Account Holder: EGAN
Account Number: 123456789
Account Balance: $100300.0
```

Challenge2.2 Exit

```python
class Player:
    def play(self):
        print("The player is playing cricket")

class Batsman(Player):
    def play(self):
        print("The batsman is batting")

class Bowler(Player):
    def play(self):
        print("The bowler is bowling")

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

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

Ln 19, Col 14 History

main.py

Run


5:05
81%
Challenge2.2
Exit
The batsman is batting
The bowler is bowling
Console
Run