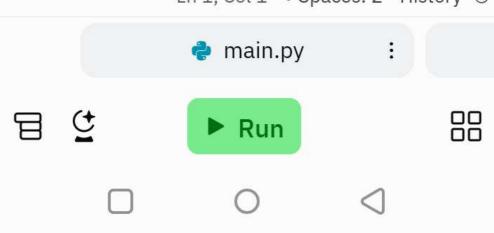
#### 11:35 4<sub>kB/s</sub> ① # Fill 2 45% 🔁 Challenge 2.2 ← Exit uei play(Seli). 7 print("The batsman is batting.") 8 9 v class Bowler(Player): def play(self): 10 \ 11 print("The bowler is bowling.") 12 13 # Create objects of Batsman and Bowler classes 14 batsman = Batsman() 15 bowler = Bowler() 16 17 # Call the play() method for each object batsman.play() 18 19 bowler.play() 20

Ln 1, Col 1 • Spaces: 2 History • main.py : Run

## Challenge 2.2

← Exit

```
1 v class Player:
      def play(self):
 2 _
 3
             print("The player is
    playing cricket.")
 4
 5 v class Batsman(Player):
6 ~
       def play(self):
 7
             print("The batsman is
    batting.")
8
9 √ class Bowler(Player):
10 \checkmark
       def play(self):
11
             print("The bowler is
    bowling.")
12
13
    # Create objects of Batsman
    and Bowler classes
14
    batsman = Batsman()
15
    bowler = Bowler()
16
17 # Call the play() method for
    each object
             Ln 1, Col 1 • Spaces: 2 History 5
               main.py
                Run
```



#### Challenge 2.1

← Exit

```
insufficient balance.")
20
    def display_balance(self):
21 ~
22
            print("Account balance
    for {} (Account #{}):
    {}".format(self.__account_holde
    r_name, self._account_number,
23
       self.___account_balance))
24
25
    # Create an instance of the
26
    BankAccount class
27
    account =
    BankAccount(account_number="123
    456789",
28
    account_holder_name="Hari
    Prabu",
29
    initial halance=5000 0)
             Ln 1, Col 1 • Spaces: 2 History 5
              main.py
                Run
```

### Challenge 2.1 ← Exit {}. New balance: {}".format(amount, self.\_\_\_account\_balance)) 11 \ else: 12 print("Invalid deposit amount. Please deposit a positive amount.") 13 def withdraw(self, amount): 14 ~ 15 <sub>~</sub> if amount > 0 and amount <= self.\_\_\_account\_balance: 16 self. account balance -= amount # Corrected this line 17 print("Withdrew {}. New balance: {}".format(amount, self.\_\_\_account\_balance)) 18 ~ else: print("Invalid 19 withdrawal amount or Ln 1, Col 1 • Spaces: 2 History 5 main.py Run

# Challenge 2.1

← Exit

```
1 v class BankAccount:
       def __init__(self,
    account_number,
    account_holder_name,
    initial balance=0.0):
3
             self._account_number =
    account_number
4
    self.__account_holder_name =
    account_holder_name
5
    self.___account_balance =
    initial_balance
6
7 _
        def deposit(self, amount):
8 ,
             if amount > 0:
9
    self.___account_balance +=
    amount
10
                 print("Deposited
    {}. New balance:
    {}".format(amount.
             Ln 1, Col 1 • Spaces: 2 History 5
              main.py
                Run
```