

File Edit Selection View Go Run ... < > Q PHYSICS 100% - x

EXPLORER ...

PHYSICS

- 1.py
- 2.py
- 3.py
- 4.py
- 5.py
- 6.py
- banking.py
- README.md
- simplebanking.py
- statement.md

banking.py > ...

```
1 import json
2 import os
3 import random
4
5 # loading acc data from file (if the file is not there I make new one)
6 def load_data():
7     if not os.path.exists("bank_data.json"):
8         return {"accs": []}
9     f = open("bank_data.json", "r")
10    dt = json.load(f)
11    f.close()
12    return dt
13
14
15 # saving the updated acc data
16 def save_data(d):
17     with open("bank_data.json", "w") as f:
18         json.dump(d, f, indent = 3)
19
20
21 # making account
22 def newAcc(d):
23     name = input("Name: ")
24     depo = float(input("Initial deposit: "))
25
26     # random acc generator (simple)
27     no = random.randint(10000,99999)
28
29     acc = {
30         "no": no,
31         "name": name,
32     }
```

Ln 121, Col 7 Spaces: 4 UTF-8 CRLF {} Python 3.13.7

File Edit Selection View Go Run ... < > Q PHYSICS 100% - x

EXPLORER ...

PHYSICS

- 1.py
- 2.py
- 3.py
- 4.py
- 5.py
- 6.py
- banking.py
- README.md
- simplebanking.py
- statement.md

banking.py > ...

```
23 def newAcc(d):
24     no = no,
25     name = name,
26     bal = depo
27
28 d["accs"].append(acc)
29 save_data(d)
30 print("Account made! Your number =", no)
31
32
33 # deposit amt
34 def deposit(d):
35     accNo = int(input("Acc no: "))
36     found = False
37
38     for a in d["accs"]:
39         if a["no"] == accNo:
40             amt = float(input("Deposit amount: "))
41             a["bal"] = a["bal"] + amt      # updated
42             found = True
43             save_data(d)
44             print("Money added.")
45             break
46
47     if not found:
48         print("No such acc.")
49
50
51 # withdrawing
```

Ln 121, Col 7 Spaces: 4 UTF-8 CRLF {} Python 3.13.7

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a folder named "PHYSICS" containing files 1.py, 2.py, 3.py, 4.py, 5.py, 6.py, banking.py, README.md, simplebanking.py, and statement.md.
- Code Editor:** The active file is "banking.py". The code implements a banking system with functions for withdrawal and checking account balances.
- Status Bar:** Displays "Ln 121, Col 7" and "Spaces: 4" along with other standard status bar information.

```
def withdraw(d):
    accNo = int(input("Enter acc no: "))
    ok = False

    for a in d["accs"]:
        if a["no"] == accNo:
            amt = float(input("Withdraw amt: "))
            if amt <= a["bal"]:
                a["bal"] -= amt
                print("Done.")
                save_data(d)
            else:
                print("Not enough bal.")
            ok = True
            break

    if ok == False:
        print("No acc found.")

# checking bal
def check(d):
    accNo = int(input("Acc number: "))
    for a in d["accs"]:
        if a["no"] == accNo:
            print("Your balance is:", a["bal"])
            return
    print("Acc not found")
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Same as the first screenshot, showing the "PHYSICS" folder.
- Terminal:** Displays the output of running the "banking.py" script. It shows a bank menu with options 1 through 5, and prompts for user input.
- Status Bar:** Displays "Ln 121, Col 7" and "Spaces: 4" along with other standard status bar information.

```
PS E:\PHYSICS> & C:/Users/Nihal/AppData/Local/Programs/Python/Python313/python.exe e:/PHYSICS/banking.py
----- BANK MENU -----
1. Create account
2. Deposit
3. Withdraw
4. Balance
5. Exit
Your choice:
```