# Case Study

Q.] Implement a banking system using OOP concepts(classes, objects, etc.) that supports operations like  
creating accounts, depositing money, withdrawing money, and checking balance.  
Design classes for different types of accounts (savings, current) using inheritance  
to encapsulate account data securely.

**Code:**

class Account:

    def \_\_init\_\_(self, number, holder, balance=0):

        self.number = number

        self.holder = holder

        self.balance = balance

    def deposit(self, amount):

        if amount > 0:

            self.balance += amount

            print(f"Deposited ₹{amount}. New balance: ₹{self.balance}")

        else:

            print("Invalid deposit amount.")

    def withdraw(self, amount):

        if amount > 0 and amount <= self.balance:

            self.balance -= amount

            print(f"Withdrew ₹{amount}. New balance: ₹{self.balance}")

        else:

            print("Invalid withdrawal or insufficient funds.")

    def check\_balance(self):

        print(f"Balance for {self.holder}: ₹{self.balance}")

class SavingsAccount(Account):

    def \_\_init\_\_(self, number, holder, balance=0, interest\_rate=0.01):

        super().\_\_init\_\_(number, holder, balance)

        self.interest\_rate = interest\_rate

    def add\_interest(self):

        interest = self.balance \* self.interest\_rate

        self.deposit(interest)

        print(f"Interest added: ₹{interest}")

class CurrentAccount(Account):

    def \_\_init\_\_(self, number, holder, balance=0, overdraft\_limit=5000):

        super().\_\_init\_\_(number, holder, balance)

        self.overdraft\_limit = overdraft\_limit

    def withdraw(self, amount):

        if amount > 0 and (self.balance - amount) >= -self.overdraft\_limit:

            self.balance -= amount

            print(f"Withdrew ₹{amount}. New balance: ₹{self.balance}")

        else:

            print("Invalid withdrawal or exceeds overdraft limit.")

# Example usage

savings = SavingsAccount("SA123", "Alice", 10000)

savings.deposit(5000)

savings.withdraw(2000)

savings.add\_interest()

savings.check\_balance()

current = CurrentAccount("CA456", "Bob", 5000)

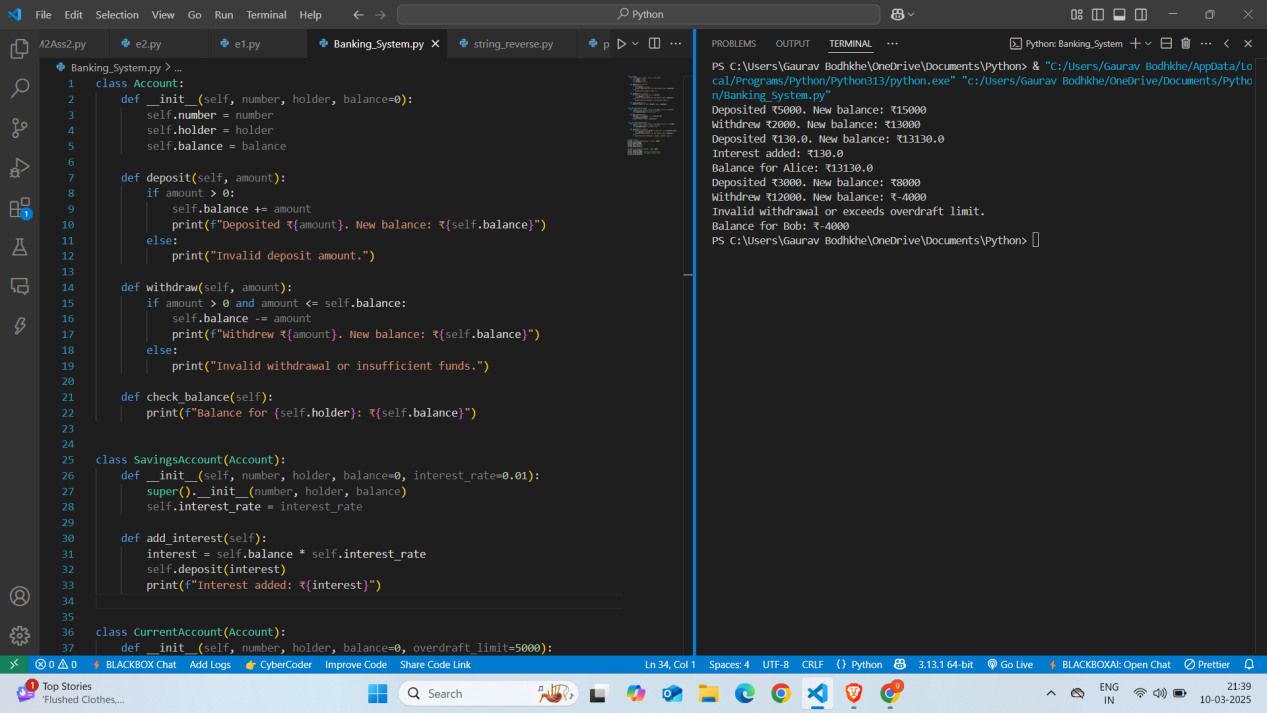
current.deposit(3000)

current.withdraw(12000)  # Within overdraft limit

current.withdraw(2000)   # Exceeds overdraft limit

current.check\_balance()

**O/P=**

****