# **Case study**

# **Gaurav\_bodkhe\_2124UCEM1041**

1. **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, acc\_num, holder, balance=0):*

*self.\_acc\_num, self.\_holder, self.\_balance = acc\_num, holder, balance*

*def deposit(self, amount):*

*if amount > 0: self.\_balance += amount; print(f"Deposited ${amount}. Balance: ${self.\_balance}")*

*else: print("Invalid deposit.")*

*def withdraw(self, amount):*

*if 0 < amount <= self.\_balance: self.\_balance -= amount; print(f"Withdrew ${amount}. Balance: ${self.\_balance}")*

*else: print("Invalid withdrawal.")*

*def check\_balance(self): print(f"Balance for {self.\_holder}: ${self.\_balance}")*

*class SavingsAccount(Account):*

*def \_\_init\_\_(self, acc\_num, holder, balance=0, interest\_rate=0.02):*

*super().\_\_init\_\_(acc\_num, 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, acc\_num, holder, balance=0, overdraft\_limit=500):*

*super().\_\_init\_\_(acc\_num, holder, balance)*

*self.\_overdraft\_limit = overdraft\_limit*

*def withdraw(self, amount):*

*if 0 < amount <= (self.\_balance + self.\_overdraft\_limit): self.\_balance -= amount; print(f"Withdrew ${amount}. Balance: ${self.\_balance}")*

*else: print("Exceeds overdraft limit.")*

*# Example*

*if \_\_name\_\_ == "\_\_main\_\_":*

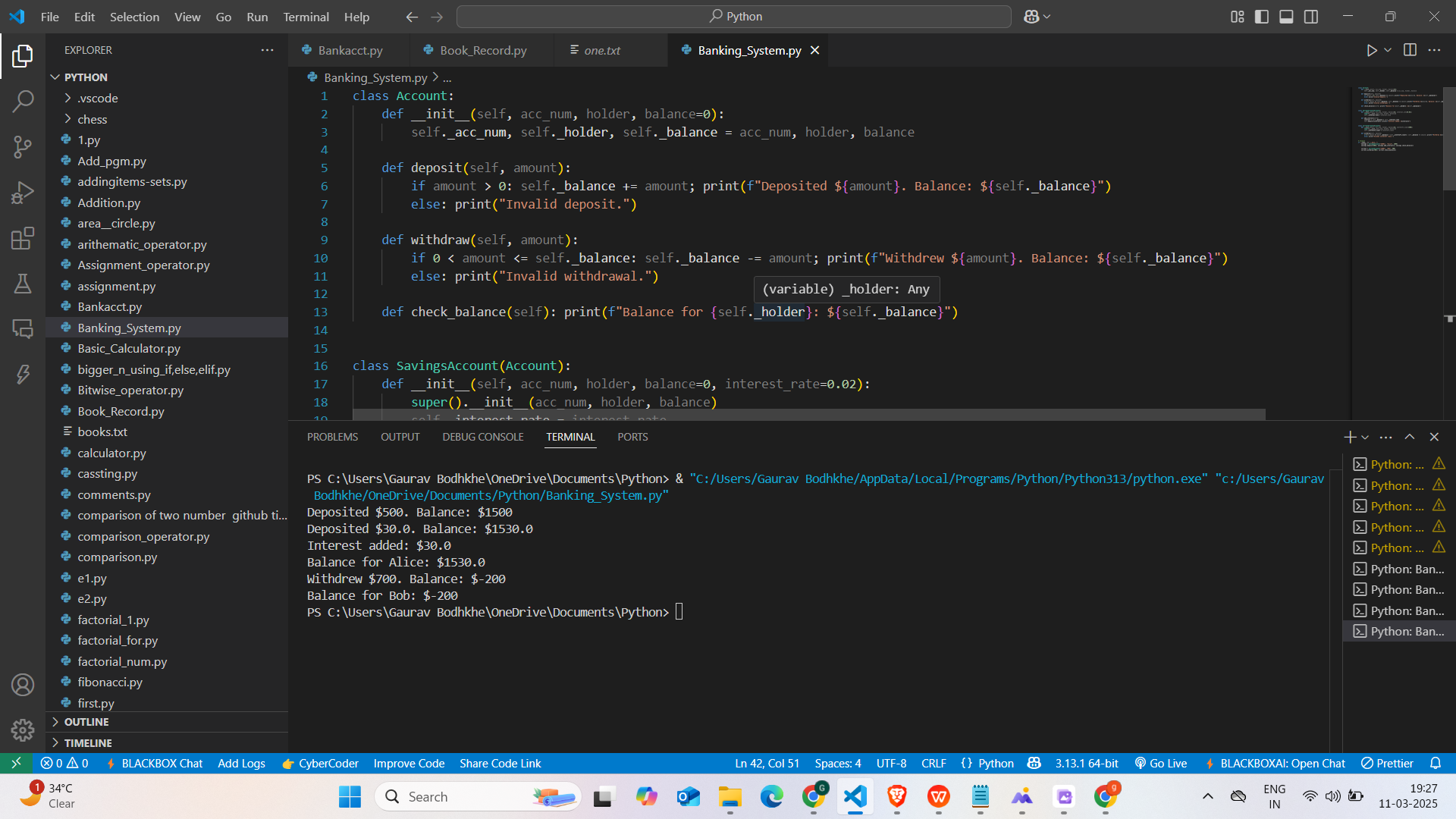
*savings = SavingsAccount("SA001", "Alice", 1000)*

*savings.deposit(500); savings.add\_interest(); savings.check\_balance()*

*current = CurrentAccount("CA001", "Bob", 500)*

*current.withdraw(700); current.check\_balance()*

**O/P=**

****