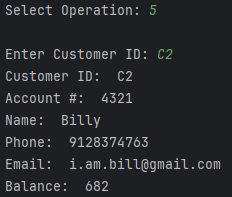
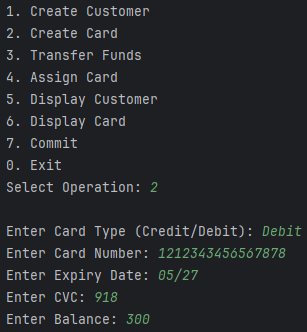
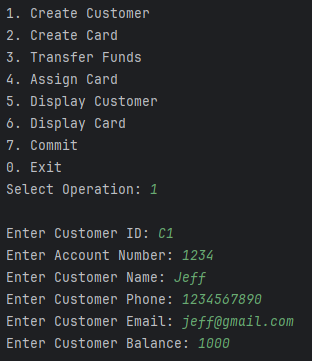
**Lab Work-4**

**Github Link:**

**Functionalities & Python Code: 7 pts**

import pickle  
class Customer:  
 def \_\_init\_\_(self):  
 self.cid = ""  
 self.acc\_no = ""  
 self.name = ""  
 self.phone = ""  
 self.email = ""  
 self.balance = ""  
 self.credit\_card = []  
 self.debit\_card = ""  
 def create\_customer(self):  
 self.cid = input("Enter Customer ID: ")  
 self.acc\_no = input("Enter Account Number: ")  
 self.name = input("Enter Customer Name: ")  
 self.phone = input("Enter Customer Phone: ")  
 self.email = input("Enter Customer Email: ")  
 self.balance = int(input("Enter Customer Balance: "))  
 def debit(self, amount):  
 print("Old Balance: ", self.balance)  
 if self.balance < amount:  
 print("Balance too low")  
 did\_not\_work = 1  
 return did\_not\_work  
 else:  
 self.balance -= amount  
 print("New Balance: ", self.balance)  
 did\_not\_work = 0  
 return did\_not\_work  
 def credit(self, amount):  
 print("Old Balance: ", self.balance)  
 self.balance += amount  
 print("New Balance: ", self.balance)  
 def display(self):  
 print("Customer ID: ", self.cid)  
 print("Account #: ", self.acc\_no)  
 print("Name: ", self.name)  
 print("Phone: ", self.phone)  
 print("Email: ", self.email)  
 print("Balance: ", self.balance)  
 def add\_card(self, card, card\_type):  
 if card\_type == "Credit":  
 self.credit\_card.append(card)  
 print(f\_card.card\_no, "added to the account of ", f\_customer.name, "as a credit card")  
 elif card\_type == "Debit":  
 self.debit\_card = card  
 print(f\_card.card\_no, "added to the account of ", f\_customer.name, "as a debit card")  
 else:  
 print("Invalid Card")  
  
class Card:  
 def \_\_init\_\_(self):  
 self.type = ""  
 self.card\_no = ""  
 self.expiry\_date = ""  
 self.cvc = ""  
 self.balance = ""  
 def create\_card(self):  
 self.type = input("Enter Card Type (Credit/Debit): ")  
 self.card\_no = int(input("Enter Card Number: "))  
 self.expiry\_date = input("Enter Expiry Date: ")  
 self.cvc = int(input("Enter CVC: "))  
 self.balance = int(input("Enter Balance: "))  
 def display(self):  
 print("Card Type: ", self.type)  
 print("Card Number: ", self.card\_no)  
 print("Expiry Date: ", self.expiry\_date)  
 print("CVC: ", self.cvc)  
 print("Balance: ", self.balance)  
  
def f\_customer(to\_find, customers\_all):  
 for find\_c in customers\_all:  
 if find\_c.cid == to\_find:  
 return find\_c  
 print("Customer not found")  
 return None  
def f\_card(to\_find, cards\_all):  
 for find\_c in cards\_all:  
 if find\_c.card\_no == to\_find:  
 return find\_c  
 print("Card not found")  
 return None  
  
customer\_file = open("bank\_customers.dat", "rb")  
all\_customers = pickle.load(customer\_file)  
customer\_file.close()  
card\_file = open("bank\_cards.dat", "rb")  
all\_cards = pickle.load(card\_file)  
card\_file.close()  
while True:  
 print("")  
 print("1. Create Customer")  
 print("2. Create Card")  
 print("3. Transfer Funds")  
 print("4. Assign Card")  
 print("5. Display Customer")  
 print("6. Display Card")  
 print("7. Commit")  
 print("0. Exit")  
 op = input("Select Operation: ")  
 print("")  
  
 #Create Customer  
 if op == "1":  
 customer = Customer()  
 customer.create\_customer()  
 all\_customers.append(customer)  
  
 #Create Card  
 elif op == "2":  
 card = Card()  
 card.create\_card()  
 all\_cards.append(card)  
  
 #Transfer Funds  
 elif op == "3":  
 paying = input("Enter Account Number for the one Paying: ")  
 paying\_c = f\_customer(paying, all\_customers)  
 paid = input("Enter Account Number for the one Being Paid: ")  
 paid\_c = f\_customer(paid, all\_customers)  
 payment = int(input("Enter Payment Amount: "))  
 failure = paying\_c.debit(paid)  
 if failure == 1:  
 print("Payment Failed")  
 else:  
 paid\_c.credit(payment)  
  
 #Assign Card  
 elif op == "4":  
 fid = input("Enter Customer ID: ")  
 f\_customer = f\_customer(fid, all\_customers)  
 fno = input("Enter Card Number: ")  
 f\_card = f\_card(fno, all\_cards)  
 f\_type = f\_card.type  
 f\_customer.add\_card(f\_card, f\_type)  
  
 #Display Customer  
 elif op == "5":  
 fid = input("Enter Customer ID: ")  
 f\_customer = f\_customer(fid, all\_customers)  
 f\_customer.display()  
  
 #Display Card  
 elif op == "6":  
 fno = input("Enter Card Number: ")  
 f\_card = f\_card(fno, all\_cards)  
 f\_card.display()  
  
 #Commit  
 elif op == "7":  
 customer\_file = open("bank\_customers.dat", "wb")  
 pickle.dump(all\_customers, customer\_file)  
 customer\_file.close()  
 card\_file = open("bank\_cards.dat", "wb")  
 pickle.dump(all\_customers, card\_file)  
 card\_file.close()  
  
 #Exit  
 elif op == "0":  
 break  
  
 else:  
 print("Invalid Operation")

**Paste the screenshots of all possible [all the operation] outputs: 3 pts**

****