from dataclasses import dataclass

GOAL: design data structures for bank accounts and customers, where accounts may be jointly managed (shared) by more than one person

```
@dataclass
class Account:
    id : int
    balance : int
@dataclass
class Customer:
    name : str
    acct : Account
# Tina is a new Customer who opens a solo account with $100
t_cust = Customer("Tina", Account(1, 100))
# Tina makes a $50 deposit
t_cust.acct.balance = t_cust.acct.balance + 50
# Maria and Jorge are new Customers who want to share an Account.
# Sharing means that either of them can make deposits or withdrawals
    and both will see the results
# Here are three possible proposals for how to set this up.
# What are the strengths/weaknesses of each?
# What do the memory diagrams look like for each?
# version 1
m_cust1 = Customer("Maria", Account(2, 250))
j_cust1 = Customer("Jorge", Account(2, 250))
# version 2
m_cust2 = Customer("Maria", Account(2, 250))
j_cust2 = Customer("Jorge", m_cust2.acct)
# version 3
new_acct = Account(2, 250)
m_cust3 = Customer("Maria", new_acct)
j_cust3 = Customer("Jorge", new_acct)
# version 4
init_bal = 250
m_cust4 = Customer("Maria", Account(2, init_bal))
j_cust4 = Customer("Jorge", Account(2, init_bal))
# Jorge wants to make a $100 deposit under version 3.
# Which of the following lines of code are appropriate?
new_acct = Account(2, new_acct.balance + 100)
new_acct.balance = new_acct.balance + 100
j_cust3.acct = Account(2, new_acct.balance + 100)
j_cust3.acct.balance = new_acct.balance + 100
j_cust3.acct.balance = j_cust3.acct.balance + 100
```

#