Week 7 OOP

Part 1 Bank Accounts

Download the Back Account File

Task 1

Add a function `deposit()` to deposit some money into your account. Sample Usage:

```
>>> myAcc = BankAccount('Alan',1000)
>>> myAcc.showBalance()
Your balance is $1000
>>> myAcc.deposit(200)
>>> myAcc.deposit(400)
>>> myAcc.showBalance()
Your balance is $1600
```

Task 2

Add a control measure when you withdraw. You must provide your name when you withdraw and it must match your name in the account

```
>>> myAcc = BankAccount('Alan',1000)
>>> myAcc.withdraw('Mary',100)
You are not authorized for this account
>>> myAcc.withdraw('Alan',10000)
Money not enough! You do not have $10000
0
>>> myAcc.withdraw('Alan',100)
100
>>> myAcc.showBalance()
Your balance is $900
```

Task 3

Add an attribute for interest rate and initialize it at the **constructor**.

Implement a function "`oneYearHasPassed()`" such that you gain the interest in your balance:

```
>>> myAcc = BankAccount('Alan',1000,0.04)
>>> myAcc.showBalance()
Your balance is $1000
>>> myAcc.oneYearHasPassed()
>>> myAcc.showBalance()
Your balance is $1040.0
>>> myAcc.oneYearHasPassed()
>>> myAcc.showBalance()
Your balance is $1081.60
```

Task 4

Define a new class of bank account called `MinimalAccount`. This class will be the same as the normal `BankAccount`, except that, if one year has passed, and your account is less than $1000, $20 dollars of administration fee will be deducted from your account. Unless the balance will be less than zero, then reset to zero. The fee will be deducted BEFORE the calculation of interest

```
>>> mySonAcc = MinimalAccount('John',40,0.04)
>>> mySonAcc.oneYearHasPassed()
>>> mySonAcc.showBalance()
Your balance is $20.8
>>> mySonAcc.oneYearHasPassed()
>>> mySonAcc.showBalance()
Your balance is $0.8320000000000007
>>> mySonAcc.oneYearHasPassed()
>>> mySonAcc.showBalance()
Your balance is $0.0
```

Further Challenge

- method `TransferTo()` in class `BankAccount`
    - Given another account, you can transfer your money to another, e.g.
    - `>>> myAcc.transferTo(myWifeAcc,500)`
- Method `setupGiro()` in class `BankAccount`. Money will be deducted every year before interest

```
>>> myAcc = BankAccount('Alan',1100,0.04)
>>> myAcc.setupGiro(40)
>>> myAcc.setupGiro(60)
>>> myAcc.oneYearHasPassed()
>>> myAcc.showBalance()
Your balance is $1040
```

- A new class `JointAccount`
    - An account has two names, anyone of them can withdraw

Part 2 Vehicles

Task 1 Add Petrol

Let's try to be more realistic that every vehicle need some petrol. Add a new method called `addPetrol(n)` that will add n liters of petrol into a vehicle. And for every "move", the vehicle will use 1 liter of petrol

```
>>> myCar.addPetrol(2)
>>> myCar.move()
Move to (0, 80)
>>> myCar.move()
Move to (0, 160)
>>> myCar.move()
Out of petrol. Cannot Move.
```

Think about ***where*** you should add this feature first.

Task 2

Implement a class `SolarTank` that does NOT need petrol. (Pulling your legs?) Just think about where you should organize your class structure.