Python HW #8

Name: Christopher Welch

1. Extending the answer from Python HW #7 for question #1, write the following python code to expand on the Car class. Include the python code you developed when answering HW #7.

A. Add class “getter” methods to return the values for model, color, and MPG

B. Add class “setter” methods to change the existing values for model, color, and MPG. The value to be used in the methods will be passed as an input argument.

C. Add class method that will calculate and return the number of miles driven, where the method is provided the size of the gas tank as an integer input

D. Below your Car class, create a new child class called mySportsCar that inherits from Car class. Make sure you call the Car class constructor with the appropriate inputs in the child class constructor.

E. Instantiate an object of the mySportsCar class. Use whatever appropriate input values that you would like.

```
#~~~~~~~~~~~~~~~Problem 1~~~~~~~~~~~~~~~~~~~~~~~~~
class Car:
    condition = "new"
    mpg = ""
    model = ""
    color = ""
    tankSize = 0
    def __init__(self, model, color, mpg):
        self.model = model
        self.color = color
        self.mpg = mpg
    def getSpecs(self):
        print("MODEL: " + self.model)
        print("COLOR: " + self.color)
        print("MPG: " + self.mpg)
    def setSpecs(self, model,color,mpg):
        self.model = model
        self.color = color
        self.mpg = mpg
    def milesDriven(self,tankSize):
        tanksDriven = float(input("\nHow many times have you had to fill the Tank?: "))
        print("With a Tank Size of: " + str(float(tankSize)) + " Gallons,")
        print("You have driven: " + str( tanksDriven * tankSize * float(self.mpg) ) + " Miles\n\n")

class mySportsCar(Car):
    def __init__(self,model,color,mpg):
        super().__init__(model,color,mpg)

my_car = Car("Corvette" , "Black" , "30")
my_car.setSpecs("370z","White","45")
my_car.getSpecs()
my_car.milesDriven(15)
newFastCar = mySportsCar("Ferrari","Red","12.3")
newFastCar.getSpecs()
```

2. In Python a class can inherit from more than one class (Java does not allow this). The resulting class will have all the methods and attributes from the parent classes. Do the following:

- Create a class called Person. In the class, define variables for storing date of birth, place of birth, and male/female attributes. In the class, define the constructor method, as well as methods for returning current values of the class attributes.

- Create a class called Employee. In the class, define variables for storing date of hire, department, and job title. In the class, define the constructor method, as well as methods for returning current values of the class attributes.

- Create a class called Salaried that inherits from both Person and Employee classes. In the class define variables for storing the salary and tax bracket. In the class, define the constructor method, as well as methods for returning current values of the class attributes.

- Instantiate a Salaried object using the following attribute default values: Date of birth: January 1, 1980; Place of birth: New York, NY; Male, Date of hire: May 1, 2005, Department: Finance; Job Title: Manager; Salary: 100000; Tax Bracket: 29%.

```
#~~~~~~~~~~~~~~~Problem 2~~~~~~~~~~~~~~~~~~~~~~~~~
class Person:
    name = ""
    dateOfBirth = ""
    placeOfBirth = ""
    sex = ""
    def __init__(self,name,dateOfBirth,placeOfBirth,sex):
        self.name = name
        self.dateOfBirth = dateOfBirth
        self.placeOfBirth = placeOfBirth
        self.sex = sex
    def identifyPerson(self):
        print("Name: " + self.name + "\nDate of Birth: " + self.dateOfBirth + "\nPlace of Birth: " +
self.placeOfBirth + "\nSex: " + self.sex)

class Employee():
    dateOfHire = ""
    Department = ""
    jobTitle = ""
    def __init__(self, dateOfHire,Department,jobTitle):
        self.dateOfHire = dateOfHire
        self.Department = Department
        self.jobTitle = jobTitle
    def identifyYourself(self):
        print("Date of Hire: " + self.dateOfHire + "\nDepartment: " + self.Department + "\nJob Title: " +
self.jobTitle)

class Salaried(Person,Employee):
    taxBracket = ""
    salary = ""
    def __init__(self,name,dateOfBirth,placeOfBirth,sex,dateOfHire,Department,jobTitle,salary,taxBracket):
        Person.__init__(self,name,dateOfBirth,placeOfBirth,sex)
        Employee.__init__(self,dateOfHire,Department,jobTitle)
        self.taxBracket = taxBracket
        self.salary = salary
    def whoAreYou(self):
        print("Name:\t\t\t"+self.name+"\nDate of Birth:\t\t"+self.dateOfBirth+"\nPlace of
Birth:\t\t"+self.placeOfBirth+"\nSex:\t\t\t"+self.sex+"\nDate of
```

```
Hire:\t\t"+self.dateOfHire+"\nDepartment:\t\t"+self.Department+"\nJob
Title:\t\t"+self.jobTitle+"\nSalary:\t\t\t"+self.salary+"\nTax Bracket:\t\t"+self.taxBracket)

christopherWelch = Salaried("Christopher Welch", "January 1, 1980", "New York, NY", "Male", "May 1, 2005",
"Finance", "Manager", "100,000", "29%")
christopherWelch.whoAreYou()
```