Python HW #7

Name: Christopher Welch

1. Write python code for the below instructions (don't forget to use the keyword "self" where appropriate in your code):

   A. Define a Class called Car

   B. Inside your Car class, define a new member variable named *condition* and give it an initial value of "new".

   C. Define the constructor to take the following inputs: model, color, and MPG. Assign these inputs to member variables of the same name.

   D. Below your Car class, create a new object named my_car that is an instance of Car, passing the following as inputs to the new object: model = "Corvette", color = "Black", and MPG = "30".

   E. At the end of your code, use a print statement to display the model of my_car, the color of my_car, the MPG of my_car, and the condition of my_car. Each of these variables should be printed on their own line in the console window

---

```
#~~~~~~Problem 1~~~~~~~~~~~~~~~~~~~~~~~~~
class Car:
    condition = "new"
    mpg = ""
    model = ""
    color = ""
    def __init__(self, model, color, mpg):
        self.model = model
        self.color = color
        self.mpg = mpg

my_car = Car("Corvette" , "Black" , "30")
print(my_car.model)
print(my_car.color)
print(my_car.mpg)
print(my_car.condition)
```

2. Write python code for the below instructions (don't forget to use the keyword "self" where appropriate in your code):

A. Define a Parent Class called Person

a. Inside your Person class, define new member variables called name, age, and gender. Initialize these variables accordingly.

b. Define the constructor to take the following inputs: name, age, and gender. Assign these inputs to member variables of the same name.

B. Define Child Class called Employee that inherits from the Person Class

a. Inside your Child class, define new member variables called title and salary. Initialize these variables accordingly.

b. Define the constructor to take the following inputs: name, age, gender, title, and salary. Assign these inputs to Class member variables of the same name. Note, you will need to specifically call the Parent constructor method inside the Child constructor, passing the appropriate inputs to the Parent constructor.

C. Instantiate an Employee object called George with the following inputs: name = "George", age = 30, gender = "Male", title = "Manager", and salary = 50000.

D. Print out the object's name, age, gender, title, and salary in **one** print statement to the console window. Your output should look like:

"George's info is: Name is George, Age is 30, Gender is Male, Title is Manager, and Salary is 50000"

```
#~~~~~~~~~~~~~~~~Problem 2~~~~~~~~~~~~~~~~~~~~~~~~~
class Person:
    name = ""
    age = ""
    gender = ""
    def __init__(self, name, age, gender):
        self.name = name
        self.age = age
        self.gender = gender

class Employee(Person):
    title = ""
    salary = ""
    def __init__(self, name, age, gender, title, salary):
        super().__init__(name,age,gender)
```

```
        self.title = title
        self.salary = salary
    def identifyYourself(self):
        print("\n\n\t" + self.name + "'s info is: Name is " + self.name + ", Age is " \
            + self.age + ", Gender is " + self.gender + ", Title is " \
            + self.title + ", and Salary is " + self.salary)

George = Employee("George", "30", "Male", "Manager", "50000")
George.identifyYourself()
```