BME 646/ ECE695DL: Homework 1

Eman Abu Ishgair

17 January 2021

1 Introduction

The assignment handles classes, inheritance, and overwriting parent class attributes.

2 Methodology

- To solve the problem, we need to create a parent class (**Countries**) and a child class (**GeoCountry**).
- The population is defined as a dictionary.
- In the child class, density is initialized to 0 since it will be calculated later
- In task 6/2nd bullet, to retrieve last count I solve the single variable equation: last_count= current_count - birth + death.
- For the user to choose a density calculating function to use, I return the function itself without the brackets ().
- In task 7 , when overwriting the function , to increase the size of the population instance variable, and change the order of last_last count in it, I used temporary local variables.
- After task 7, any future calculations for density should give different results, because the overwritten net_population has a different equation for current_net. To accommodate this, I changed the density calculators 1,2 to use the parent net_population if the size of the instance variable population is 3, and use the overwritten net_population if the size of the instance variable population is 4.
- If the size of the instance variable population is 4, when calling density calculator 2 , the function does retrieve the last count using a different equation, which is the new equation for the current net.
- If the net_population attribute of GeoCountry is being called with the instance variable population of size 4 , the population size should not be changed, to do that I had condition on the size of population.

3 Implementation and Results

```
class Countries():

   def __init__(self,capital,population):

       self.capital=capital

       self.population= {}
```

BME 646/ ECE695DL: Homework 1

Eman Abu Ishgair

17 January 2021

```
        self.population["birth"]=population[0]
        self.population["death"]=population[1]
        self.population["last_count"]=population[2]



    def net_population(self):
                                                                    current_net=
self.population["birth"]-self.population["death"]+self.population["last_count"]
        return current_net


class GeoCountry(Countries):
    def __init__(self,capital,population,  area):
        Countries.__init__(self,capital,population)
        self.area=area
        self.density=0


    def density_calculator1(self):
        if (len(self.population)==3):
            self.density= Countries.net_population(self)/ self.area
        else:
            self.density= self.net_population()/ self.area
```

BME 646/ ECE695DL: Homework 1

Eman Abu Ishgair

17 January 2021

```
   def density_calculator2(self):
       if (len(self.population)==3):

self.population["last_count"]=self.population["last_count"]-self.population["birth"
]+self.population["death"]
           self.density= Countries.net_population(self)/ self.area
       else:

self.population["last_count"]=2*(self.population["last_count"]-self.population["bir
th"]+self.population["death"])- self.population["second_last_count"]
           self.density= self.net_population()/ self.area




   def net_population(self):
       if(len(self.population)==3):
           temp1=Countries.net_population(self)
           temp2=self.population["last_count"]
           self.population.pop("last_count")
           self.population["second_last_count"]=temp2
           self.population["last_count"]=temp1
```

BME 646/ ECE695DL: Homework 1

Eman Abu Ishgair

17 January 2021

```
                current_net=self.population["birth"]  -  self.population["death"]  +
(self.population["second_last_count"] + self.population["last_count"]) / 2

            return current_net

        else:

                current_net=self.population["birth"]  -  self.population["death"]  +
(self.population["second_last_count"] + self.population["last_count"]) / 2

            return current_net




    def net_density(self,choice):

        if(choice==1):

            return self.density_calculator1

        if (choice==2):

            return self.density_calculator2







if __name__ == "__main__":

    a_country= Countries("Piplipol",[ 40,30,20])

    a_geoCountry=GeoCountry("Polpip", [ 55,10,70], 230)
```

BME 646/ ECE695DL: Homework 1

Eman Abu Ishgair

17 January 2021

4 Lessons Learned

I had some errors defining instance variables, I tried to define them as attributes but it didn't work.I learned that Instance variables are different from class attributes , Instance variables are owned by the specific instances of a class, for two different class instances, the instance variables values are different. Class attributes are defined directly in the class and shared by all class instances, while instance variables are defined in the constructor.

5 Suggested Enhancements

The tasks are explained well, and the outcomes are made clear, so I have no suggested enhancements.