Documentation of Steps

**Step 1: Informal list of responsibilities of objects.**

These are the responsibilities associated with this task:

* Countries – The names are needed.
* Area – The area (float number) is needed.
* Population – The population (integer number) is needed.
* Population density – This one is a hidden responsibility as it is needed for the task but it is dependent on population and density which we must calculate ourselves.

**Step 2: Specifying the public interface.**

There is seven methods in total for this class.

* def populationDensity(self):, returns the population density.

Methods that can handle list as input:

* def list\_country\_w\_largest\_area(countrylist):, returns the country with the largest area.
* def list\_country\_w\_largest\_pop(countrylist):, returns the country with the largest population.
* def list\_country\_w\_largest\_popdensity(countrylist):, returns the country with the largest population density.

Methods that can handle dictionaries as input

* def dict\_country\_w\_largest\_area(countrydict):, returns the country with the largest area
* def dict\_country\_w\_largest\_pop(countrydict):, returns the country with the largest population.
* def dict\_country\_w\_largest\_popdensity(countrydict):, returns the country with the largest population density.

**Step 3: Document the public interface.**

I document most of the public interface but not all in detail since the methods are so similar.

A screenshot of a computer program

Description automatically generated

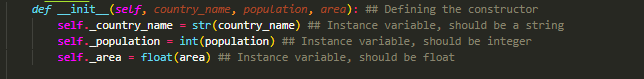
**Step 4: Determine instance variables.**

The instance variables are:

* \_country\_name (Name of the country).
* \_population (A countrys population).
* \_area (A countrys total land area)

The instance variables are protected using (\_) to indicate that they should not be used outside the class.

**Step 5: Implement the constructor.**

The constructor is implemented as follows:  


**Step 6: Implement the methods.**

The methods are implemented as follows:

A screenshot of a computer program

Description automatically generated

**Step 7: Test the class.**

To test the code I run it from the terminal using a simple example, where the largest country should be country2 with area = 10000, the largest in population should be country1 with population = 2000, and the country with highest density should be country3 with density = 1. And as we can see that is the result we get:  
A screenshot of a computer program

Description automatically generated