CS2121B - Assignment 1 - 2020

Python Classes: Design and Implementation

Published: January 13th, 2020

Due date: January 27th, 2020 at 11:00 pm via OWL

Note Well:

You must take note of the following:

- You are given two (2) .py files that you are to complete. For your work with these files take note of the following:
  - Your function names, along with the parameters they take **must** remain the same; **you cannot alter these**. If you alter these, you will lose significant marks and risk getting a mark of zero.
  - A file called `test.py` will be used by the TA to mark your assignment – I have provided you with the file we will actually use to test your code. You are free to use this on your own code.
  - Please **do not** prompt the user for input; all functions will be called by just calling the functions.
- You need to comment your code so that someone familiar with Python, but who has not read your code, can easily understand what each function does.
- If there is something you are not sure how to do, you should first Google it or look it up in the Python documentation online: `https://docs.python.org/3/index.html`

Purpose

The purpose of this assignment is to become familiar with writing your own classes, using them, and becoming familiar with collections.

Summary

You are to write a program that will keep a listing of company objects. These company objects will store some information about companies. You will write the `Company` class and the `Listing` class.[1]

[1]The data used for this assignment is taken from the OpenData500 Global Network website at `http://www.opendata500.com/`. The precise number of employees are not listed in this database, so we have chosen a value in the value range indicated in the database.

Provided:

You are provided with a testing script (`test.py`), two incomplete python scripts (`Company.py` and `Listing.py`) you are to finish, and two .txt files (`countries.txt` and `companies.txt`) to be read into your program (do not change these).

`Company` class

The `Company` class will describe a company. This class will have five (5) attributes: `name`, `category`, `country`, `num_employees`, and `year_founded`. The constructor is already written for you. You will need to complete four (4) functions:

1. `__str__`
   - This function will return a string in the format of:
     `name + ' in ' + category + ' sector based in ' + country`
     Ex: `Wolfram Research in Data/Technology sector based in United States`
2. `getAge`
   - This function will return the number of years since the company was founded.
3. `save`
   - This function will take a `fname` as parameter, and will save a single line string to the file in the format of: `name|category|country|num_employees|year_founded`
     Ex. `Wolfram Research|Data/Technology|United States|800|1987`
   - This function should append this line to the last line of the file, and create the file if it does not already exist.
4. `__eq__`
   - This function will tell us if two companies are the same. Our requirement for companies being the same will just check if the names are the same.

`Listing` class

The `Listing` class will will hold a collection of companies for us. It will maintain a list of companies and a dictionary of countries. This class will also have a few methods you are to implement. You will need to complete 11 functions:

1. `__init__`
   - Complete the constructor.

- The constructor takes a country file name, containing company names and their corresponding countries, and a company file name with company details.
- The constructor will read in the information from the files with the names provided to the constructor and store them in the dictionary and list.
- The dictionary will hold the country information from the first (country) file.
- The list will store all the company information from the second (company) file.
- The constructor will create the Company objects to be stored in the list based on the information in the second file.

2. `__iter__`
   - Complete the code to return an iterator for the company listing.
3. `__next__`
   - Complete the code to iterate through the company list.
4. `__str__`
   - Complete the function to return a string where each company in the list will be printed on its own line:
     Ex:
     ```
     Wolfram Research in Data/Technology sector based in United States
     visioninside in Consulting sector based in Korea
     ...
     ```
5. `addCountry`
   - This function will take a company name and country name, and add the pair to the country dictionary (of countries corresponding to particular companies).
6. `addCompany`
   - This method will take a name, category, number of employees, and founding year, (but not a country), and attempt to add a new company with the provided information to the list. Do not attempt to sort this list.
   - We do not want duplicates, so if the company is already in the list (based on name), you will raise an exception (specifically, `ValueError`). This error will have the message
     `This company is already in the listing.`
   - If we do not know what country the company is in (company is not in the country dictionary), you will raise an exception (specifically, `ValueError`). This error will have the message
     `Country information unavailable for this company.`
     `Try first adding this company and country to the country list of this listing.`

7. `removeCompany`
   - This function will take a company name and attempt to remove the company with the provided name from the list.
   - If the company is not actually in the list, you will raise an exception (`ValueError`). This error will have the message
     `Company not in list. Nothing was removed.`
8. `companyDetails`
   - This function will take a company name and will return a reference to the company with the matching name.
   - If the company does not exist, simply return a message saying: `cname + ' is not in the listing.'`
9. `filterByCountry`
   - This function will take a country name and print out all the details for each company in the list that is in that country.
   - Ex:
     `Wolfram Research in Data/Technology sector based in United States`
     `Forrester Research in Research & Consulting sector based in United States`
     `...`
10. `saveListing`
    - This function will take a string for an output file name
    - This function will create an output file where we will save some details about the companies in the list.
    - If a file with that name already exists in the folder, you should overwrite that file.
    - HINT: use each company's save function
11. `largestCompany`
    - This function will take a country as an input argument and find the company with the largest number of employees in that country and will return some details in this format:
      `'companyName is the largest company in ' + countryName.`
      `It has grown to a size of ' + numberOfEmployees + ' employees in ' + ageOfCompany + ' years.'`
      Ex:
      `Wolfram Research is the largest company in United States.`
      `It has grown to a size of 800 employees in 32 years.`

Submitting the Assignment

You will submit the file via OWL, under the assignments tab. **Only submit your complete python files** (`Company.py` and `Listing.py`) via OWL. Please follow the instructions carefully and do what is asked, but not more, to ensure that you do not unnecessarily lose marks. Make sure not to modify the function names and the .txt files provided.