CIS 41B - Lab 1: Review of CIS 41A topics; New topics: iterables, callables

Write an application that lets the user look up the top world universities.

**Input file**

The input file is uniRanks.csv ([source](https://www.timeshighereducation.com/world-university-rankings/2024/world-ranking)), which contains the 2024 ranking of universities around the world.

Here is the sample first line of the file:

1,University of Oxford,98.5,United Kingdom,21750,10.9,"Geography,Chemistry,Chemical Engineering,Biological Sciences,General Engineering,Computer Science,Art, Performing Arts & Design,Communication & Media Studies,Electrical & Electronic Engineering,Law,Economics & Econometrics,Geology, Environmental, Earth & Marine Sciences,Politics & International Studies (incl Development Studies),History, Philosophy & Theology,Medicine & Dentistry,Languages, Literature & Linguistics,Business & Management,Accounting & Finance,Mechanical & Aerospace Engineering,Mathematics & Statistics,Physics & Astronomy,Psychology,Other Health,Archaeology,Sociology,Civil Engineering,Education"

The columns are: rank, name, ranking score, country, number of students, ratio of students/staff, “list of top academic fields taught at the university”

**Overview**

Lab 1 consists of 2 classes that are related and a main block.

The 2 classes are Uni (for university) and UniRank (for university ranking)

* The Uni class represents one university and stores data for each university.
* The UniRank class has a container of Uni objects and provides searches for the universities.

The user has 3 ways to look up the universities:

1. Show all universities by ranking order, which is the same order as they appear in the file.
2. Show all universities for one country.
3. Show all universities sorted by number of students or sorted by ratio of students/staff

The file lab1.py is already started for you. It is recommended that you add code in the order shown below.

**The Uni class**

The header of the Uni class is already in lab1.py. Add code to do the following:

1. The Uni class contains the following 6 data values for a university: rank, name, country, number of students, ratio of students/staff, first 3 academic fields (top specialties of the university)
2. Any getter methods to allow access to the university data.   
   Use the property decorator with these getter methods to make them more user friendly.
3. A special method (double underscore) that returns the string to identify the university. The returned string should be in a 3-line format: rank name: country  
    students: num\_of\_students, ratio: students/staff ratio  
    top fields: the 3 comma separated academic fields

where the gray text are actual data values and the purple text are actual text strings. See sample output.

**The UniRank class**

The header of the UniRank class is already in lab1.py. Add code to do the following:

1. Create a class attribute to store the input file name uniRanks.csv.
2. The \_\_init\_\_ method accepts a number from the user, which is the number of universities that the user wants to work with. Here are the tasks for the \_\_init\_\_ method:

* If the input number less than 1, a ValueError exception should occur with an error message to let the user know the number has to be at least 1.
* Use a generator (see item 3) to get from file the number of universities that the user requests, or up to the max number of available universities.
* For each university that’s received from the generator, create a Uni object to store the university data, then store the Uni object into a container of your choice.
* When all the appropriate number of universities are read in, print:
  + The number of universities that were read in.
  + An optional message to say that there are only N universities, if the user input number is larger than N.   
    Your code should not have to increment a count to get N, or hard code N, let the generator tell you).
  + The number of universities for each country in the container, sorted by number of universities.

See sample output.

1. A generator that accepts an input number, which is the number of lines that the caller wants from the file.
   * The generator reads one line at a time from the file and yields each valid line, until the number of lines is reached or until the input line is not valid, whichever comes first.

* A line from the file is not valid if the first value (the rank) is not an integer. (This happens on line 202, so you can take a look at it, but your code should not count on the 202 value or any constant. When I test your code, the first invalid line may not be line 202.)
* Don’t forget that a generator either yields data or raises StopIteration. A generator doesn’t yield False or empty string or any value that’s not actual data.
* Since this is a generator, don’t create temporary containers to do the work.

1. A method that returns the container of universities.
2. A method that accepts a country name as input argument, and returns a container of all Uni objects with that country name (or all universities in that country).

* Use the filter() function to filter out all universities that don’t have that country name.

1. A method that accepts a sort criterion and a number N (of universities) as input arguments.
   * The sort criterion is either “students” or “ratio”
   * The input number N is the number of universities that the user wants to see in the output.  
     If the N is not at least 1, a ValueError exception should occur with an error message to let the user know that the number needs to be at least 1.
   * Use the sorted function with the appropriate “key” argument to sort the Uni objects either by students or by ratio of students/staff. You should not have to use any intermediate container to do the sorting.
   * The sort is in descending order: highest value first, lowest value last.
   * Return a container of N sorted Uni objects.

[1pt EC challenge: call the sorted function only one time in the method. This means don’t use an if statement to call sorted by student or call sorted by ratio]

1. Add any support methods as needed.

**Decorator**

Outside both classes, create a decorator that works with any function that returns a container.

* The decorator prints the size of the container, which is the return value of the function that it decorates.
* Use the decorator with methods 4,5,6 of the UniRank class above.

**The main block**

The main block is already started in lab1.py to create a UniRank object with 15 countries. Feel free to change the input number to test your code.

Add code to do the following:

1. Open the testcases.csv file and read in each line of the file. Each line is a test case for the UniRank class.

The data in the testcases.csv are:

Lines in the file Equivalent calls to the UniRank object methods

|  |  |
| --- | --- |
| a  c,switzerland  c,united kingdom  c,south korea  s,students,3  s,ratio,100 | ‘a’ for all: get the container of all the universities  ‘c’ for country: get a container of all universities in ‘switzerland’  ‘c’ for country: get a container of all universities in ‘united kingdom’  ‘c’ for country: get a container of all universities in ‘south korea’  ‘s’ for sort: get a container of 3 universities, sorted by students  ‘s’ for sort: get a container of 100 universities, sorted by ratio |

1. With each line of text that you read in, call the appropriate UniRank method and pass to it the input arguments from the line.

* You cannot use an if statement, case statement, a loop that runs one time, or a try except to choose the appropriate method to call. Instead, take advantage of functions being first class objects in Python and create a look up table for the methods.

1. Each of the 3 UniRank methods returns a container.
   * Before printing the container, print the line of text from the file.
   * Then:  
     - either print all the Uni objects in the container, take advantage of the special method in the Uni class to make it easy to print each university data  
     - or print “no data available”

**Documentation**

- At the top of the file: put your name and lab number, at minimum. Additional documentation is always okay.

- For each public method: add a docstring

You don't need to add docstrings for private method or 'get' method where the method only returns a data attribute, but you're welcome to do so.

**Test**

Sample output from the testcases.csv file and existing code in lab1.py.  
Don’t forget to change the testcases.csv file and the input number to UniRank object to thoroughly test your code. I will not be using the same testcases.csv file or the number 15 when I test your code.

Number of universities read in: 15

Number of universities for each country:

United States: 9

United Kingdom: 3 # this block of output is from \_\_init\_\_ method

China: 2

Switzerland: 1

Test case: a

from decorator: returning container of size 15

1 University of Oxford, United Kingdom:

students: 21750, ratio: 10.9

top fields: Geography, Chemistry, Chemical Engineering

2 Stanford University, United States:

students: 14517, ratio: 6.4

top fields: Computer Science, Communication & Media Studies, Electrical & Electronic Engineering

3 Massachusetts Institute of Technology, United States:

students: 11085, ratio: 8.0

top fields: Architecture, Economics & Econometrics, Archaeology

4 Harvard University, United States:

students: 20050, ratio: 9.0

top fields: Sociology, Architecture, Physics & Astronomy

5 University of Cambridge, United Kingdom:

students: 20565, ratio: 11.5

top fields: Politics & International Studies (incl Development Studies), Biological Sciences, Business & Management

6 Princeton University, United States:

students: 7753, ratio: 7.3

top fields: History, Philosophy & Theology, Agriculture & Forestry

7 California Institute of Technology, United States:

students: 2240, ratio: 6.1

top fields: Computer Science, Chemistry, Chemical Engineering

8 Imperial College London, United Kingdom:

students: 20275, ratio: 11.8

top fields: Computer Science, Mechanical & Aerospace Engineering, Chemical Engineering

9 University of California, Berkeley, United States:

students: 39991, ratio: 17.9

top fields: Chemistry, Electrical & Electronic Engineering, Geology

10 Yale University, United States:

students: 11924, ratio: 5.2

top fields: Law, Computer Science, Electrical & Electronic Engineering

11 ETH Zurich, Switzerland:

students: 22619, ratio: 15.4

top fields: Mechanical & Aerospace Engineering, Education, Agriculture & Forestry

12 Tsinghua University, China:

students: 38518, ratio: 11.2

top fields: Mechanical & Aerospace Engineering, Sociology, Architecture

13 The University of Chicago, United States:

students: 15792, ratio: 6.2

top fields: Chemistry, Languages, Literature & Linguistics

14 Peking University, China:

students: 33064, ratio: 10.9

top fields: History, Philosophy & Theology, Geology

15 Johns Hopkins University, United States:

students: 15772, ratio: 4.1

top fields: Civil Engineering, Geology, Environmental

Test case: c, switzerland

from decorator: returning container of size 1

11 ETH Zurich, Switzerland:

students: 22619, ratio: 15.4

top fields: Mechanical & Aerospace Engineering, Education, Agriculture & Forestry

Test case: c, united kingdom

from decorator: returning container of size 3

1 University of Oxford, United Kingdom:

students: 21750, ratio: 10.9

top fields: Geography, Chemistry, Chemical Engineering

5 University of Cambridge, United Kingdom:

students: 20565, ratio: 11.5

top fields: Politics & International Studies (incl Development Studies), Biological Sciences, Business & Management

8 Imperial College London, United Kingdom:

students: 20275, ratio: 11.8

top fields: Computer Science, Mechanical & Aerospace Engineering, Chemical Engineering

Test case: c, south korea

from decorator: returning container of size 0

no data available

Test case: s, students, 3

from decorator: returning container of size 3

9 University of California, Berkeley, United States:

students: 39991, ratio: 17.9

top fields: Chemistry, Electrical & Electronic Engineering, Geology

12 Tsinghua University, China:

students: 38518, ratio: 11.2

top fields: Mechanical & Aerospace Engineering, Sociology, Architecture

14 Peking University, China:

students: 33064, ratio: 10.9

top fields: History, Philosophy & Theology, Geology

Test case: s, ratio, 100

from decorator: returning container of size 15

9 University of California, Berkeley, United States:

students: 39991, ratio: 17.9

top fields: Chemistry, Electrical & Electronic Engineering, Geology

11 ETH Zurich, Switzerland:

students: 22619, ratio: 15.4

top fields: Mechanical & Aerospace Engineering, Education, Agriculture & Forestry

8 Imperial College London, United Kingdom:

students: 20275, ratio: 11.8

top fields: Computer Science, Mechanical & Aerospace Engineering, Chemical Engineering

5 University of Cambridge, United Kingdom:

students: 20565, ratio: 11.5

top fields: Politics & International Studies (incl Development Studies), Biological Sciences, Business & Management

12 Tsinghua University, China:

students: 38518, ratio: 11.2

top fields: Mechanical & Aerospace Engineering, Sociology, Architecture

1 University of Oxford, United Kingdom:

students: 21750, ratio: 10.9

top fields: Geography, Chemistry, Chemical Engineering

14 Peking University, China:

students: 33064, ratio: 10.9

top fields: History, Philosophy & Theology, Geology

4 Harvard University, United States:

students: 20050, ratio: 9.0

top fields: Sociology, Architecture, Physics & Astronomy

3 Massachusetts Institute of Technology, United States:

students: 11085, ratio: 8.0

top fields: Architecture, Economics & Econometrics, Archaeology

6 Princeton University, United States:

students: 7753, ratio: 7.3

top fields: History, Philosophy & Theology, Agriculture & Forestry

2 Stanford University, United States:

students: 14517, ratio: 6.4

top fields: Computer Science, Communication & Media Studies, Electrical & Electronic Engineering

13 The University of Chicago, United States:

students: 15792, ratio: 6.2

top fields: Chemistry, Languages, Literature & Linguistics

7 California Institute of Technology, United States:

students: 2240, ratio: 6.1

top fields: Computer Science, Chemistry, Chemical Engineering

10 Yale University, United States:

students: 11924, ratio: 5.2

top fields: Law, Computer Science, Electrical & Electronic Engineering

15 Johns Hopkins University, United States:

students: 15772, ratio: 4.1

top fields: Civil Engineering, Geology, Environmental

Partial output when the UniRank object is created as:

r = UniRank(300)

Number of universities read in: 202 (only 202 available out of the 300 requested)

Number of universities for each country:

United States: 56

United Kingdom: 25

Germany: 21

China: 13

Australia: 11

Netherlands: 11

Canada: 8

Switzerland: 7

Sweden: 6

South Korea: 6

Japan: 5

... and other countries ...