

Introduction

Postsecondary education is becoming popular around the world with students trying to find best choices when selecting their future university. Universities usually participate in the global ranking systems which sort the participating universities based on certain criteria. This ranking helps institutions build global brand visibility, forge strategic partnerships and assists in recruiting international talents. In this assignment, you are provided with some data about universities rankings and you are asked to apply your knowledge through the course so far to extract some information.

- Before starting coding this assignment, read all the instructions carefully as it guides you to accurately write your outputs.
- Try to read the [Tips and Guidelines](#) section to have a good start with your code.

In this assignment, you will get practice with:

- Using functions
- Complex data structures
- Text processing
- File input and output
- Exceptions in Python
- Using Python modules
- Adhering to specifications by testing programs and developing test cases.
- Writing code that is used by other programs.

Files

For this assignment, you will be **provided with** three files: **TopUni.csv**, **capitals.csv**, and **tester.py**.

You are asked to create one Python file: **univRanking.py**.

TopUni.csv

This file contains data about the universities ranking nationally and internationally. The data is stored in a comma separated file (.CSV). Each line within the file contains **9** fields as follows:

- World Rank (International Rank) --> **Used**: This field represents the ranking of the university among all other universities in the world. Smaller values of this field means a higher rank i.e., the top rank is 1, the second rank is 2 and so on.
- Institution name --> **Used**: This field represents the name of the university.
- Country --> **Used**: This field represents the country where the university is located at.
- National Rank --> **Used**: This field represents the ranking of the university among all other universities within the same country.
- Quality of Education Rank --> **NOT Used**
- Alumni Employment Rank --> **NOT Used**
- Quality of Faculty Rank --> **NOT Used**
- Research Performance Rank --> **NOT Used**
- Score --> **Used**: This field represents an overall score of the university. This score reflects quality of the education, alumni employment rate, etc.

The following is a sample data snapshot:

World Rank ▲▼	Institution ▼	Country ▼	National Rank ▼	Quality of Education Rank ▼	Alumni Employment Rank ▼	Quality of Faculty Rank ▼	Research Performance Rank ▼	Score ▼
1	Harvard Universi	USA	1	1	1	1	1	100
2	Massachusetts I	USA	2	4	12	2	8	96.7
3	Stanford Univers	USA	3	10	4	3	2	95.1
4	University of Cai	United Kingdo	1	3	25	4	10	94.1

capitals.csv

This file contains geographical data for several countries around the world. The data is stored in a comma separated file (.CSV). Each line within the file contains **6** fields as the following:

- Country Name --> **Used**: This field represents the country name.
- Capital --> **Used**: This field represents the capital of the country.
- Latitude --> **NOT Used**
- Longitude --> **NOT Used**
- Country Code --> **NOT Used**
- Continent --> **Used**: This field represents the continent where the country is located at.

The following is a sample data snapshot:

Country Name ▲	Capital ▼	Latitude ▼	Longitude ▼	Country Code ▼	Continent ▼
Afghanistan	Kabul	34.52	69.18	AF	Asia
Aland Islands	Mariehamn	60.12	19.9	AX	Europe
Albania	Tirana	41.32	19.82	AL	Europe
Algeria	Algiers	36.75	3.05	DZ	Africa
American Samoa	Pago Pago	-14.27	-170.7	AS	Australia
Andorra	Andorra la Vel	42.5	1.52	AD	Europe

univRanking.py

This file should contain all your work. In this file you need to write a set of functions that will help you complete the requirement of this assignment. **The main objective is to read the content of the two .csv files and display information extracted from these files.** This information is used to provide general insights on the countries, continents, universities, and more specific information about a country determined by the user. **All the required information should be written out to a file named `output.txt`**

The following is the required information that should be presented within `output.txt` in order:

1. Universities count

You need to show the total number of the universities in the `TopUni.csv` file. The output should be *Total number of universities => \$\$\$* where the \$\$\$ should be replaced by the total number of the universities. The following is an example output:

```
≡ output.txt
```

```
Total number of universities => 2000
```

2. Available countries

You need to show the list of all country names available in the **TopUni.csv** file in the order that they appear in the file. Each country name should be displayed only once without repetition. The names should be in upper case. Country names are separated by a comma and a space after it. The output should be as follows: *Available countries => COUNTRY1, COUNTRY2, COUNTRY3*, where COUNTRY1, COUNTRY2, COUNTRY3, should be replaced by the list of all country names. If the printed list of countries end with a comma and a space, that is fine. The following is an example output:

```
≡ output.txt
```

```
Total number of universities is 2000
```

```
Available countries => USA, UNITED KINGDOM, JAPAN, FRANCE, CANADA, SWITZERLAND, SI
```

3. Available continents

You need to show the list of all continent names available in the **capitals.csv** file. You must display the continents names corresponding to the countries listed in the previous point. For example, if the list of countries displayed in the previous point is CANADA, JORDAN, this means that the list of continents should be NORTH AMERICA, ASIA. Each continent name should be displayed only once without repetition. The names should be in upper case. Every two names are separated by a comma and a space after it (,). The output should be like *Available continents => CONTINENT1, CONTINENT2, CONTINENT3*, where CONTINENT1, CONTINENT2, CONTINENT3, should be replaced by the list of all continents names. The following is an example output:

```
≡ output.txt
```

```
Total number of universities is 2000
```

```
Available countries => USA, UNITED KINGDOM, JAPAN, FRANCE, CANADA, SWITZERLAND, SI
```

```
Available continents => NORTH AMERICA, EUROPE, ASIA, AUSTRALIA, SOUTH AMERICA, AFF
```

The following requirements requires the user to specify a country name to display the corresponding information. We will refer to this country as the term Selected Country

4. The university with top international rank

You need to show the **world rank** and the **name** of the university that has the highest international rank within the selected country. This means that you need to check the international rank of all universities in the selected country and show the required information. The output should be as follows: *At international rank => \$\$\$ the university name is => UNIVERSITY* where \$\$\$ should be replaced by the international rank number, and UNIVERSITY should be replaced by the university name. The following is an example

output:

≡ output.txt

Total number of universities = 200

Universities sorted by CS: MIT, MIT, HARVARD, JHU, CMU, UIUC, STANFORD, UC

Universities sorted by CS: MIT, MIT, HARVARD, JHU, CMU, UIUC, STANFORD, UC

At international rank => 31 the university name is => SEOUL NATIONAL UNIVERSITY

5. The university with top national rank

You need to show both the **national rank** and the **name** of the university that has the highest national rank within the selected country. This means that you need to check the national rank of all universities in the selected country and show the required information. The output should be like *At national rank => \$\$\$ the university name is => UNIVERSITY* where \$\$\$ should be replaced by the national rank number, and UNIVERSITY should be replaced by the university name. The following is an example output:

≡ output.txt

Total number of universities = 200

Universities sorted by CS: MIT, MIT, HARVARD, JHU, CMU, UIUC, STANFORD, UC

Universities sorted by CS: MIT, MIT, HARVARD, JHU, CMU, UIUC, STANFORD, UC

At international rank => 31 the university name is => SEOUL NATIONAL UNIVERSITY

At national rank => 1 the university name is => UNIVERSITY OF TORONTO

6. The average score

You need to show the average score of **all** universities within the selected country. The average score is calculated according to the following equation:

$$\text{Average score} = \frac{\sum \text{of all university scores within the selected country}}{\text{Number of universities within the selected country}}$$

The average score value should be rounded to two decimal places. The output should be as follows: *The average score => \$\$\$%* where \$\$\$ should be replaced by the average score and a percentage symbol. The following is an example

output:

≡ output.txt

Total number of universities = 2000

Available countries = USA, NORTH AMERICA, JAPAN, FRANCE, CANADA, AUSTRALIA, 1

Available continents = NORTH AMERICA, CANADA, ASIA, AUSTRALIA, SOUTH AMERICA, 4

US universities rank = 20 the university rank is = UNIVERSITY OF TORONTO

US national rank = 1 the university rank is = UNIVERSITY OF TORONTO

The average score => 73.82%

7. The continent relative score

The relative score is defined as the ratio between the average score (calculated in the previous point) divided by the highest score within the continent where the selected university is located. This is defined as:

$$\text{Relative score} = \frac{\text{The average score}}{\text{The highest score within the continent where the selected university is located}}$$

This means that you need to check the scores of all universities in all countries that belong to the continent of the selected country. The relative score value should be rounded to two decimal places. The output should be like *The relative score to the top university in CONTINENT is => (\$\$\$ / &&&) x 100% = @@@%* where CONTINENT is the continent of the selected country, \$\$\$ should be replaced by the average score, &&& should be replaced by the highest score in the continent, @@@ should be replaced by the relative score and a percentage symbol. The following is an example output:

≡ output.txt

Total number of universities = 2000

Available countries = USA, NORTH AMERICA, JAPAN, FRANCE, CANADA, AUSTRALIA, SOUTH AMERICA, NORTH AMERICA, 10 AR

Available continents = NORTH AMERICA, CANADA, ASIA, AUSTRALIA, SOUTH AMERICA, AFRICA, CENTRAL AMERICA,

US universities rank = 20 the university rank is = UNIVERSITY OF TORONTO

US national rank = 1 the university rank is = UNIVERSITY OF TORONTO

The average score = 73.82%

The relative score to the top university in NORTH AMERICA is => (77.72 / 100.0) x 100% = 77.72%

8. The capital city

You need to show the capital of the selected country. For example, if the selected country is CANADA, the capital should be OTTAWA. The output should be as follows: *The capital is => CAPITAL* where CAPITAL is the capital name of the selected country. The following is an example output:

≡ output.txt

Total number of universities = 2000

Available countries = USA, UNITED KINGDOM, JAPAN, FRANCE, CANADA, AUSTRALIA, SOUTH AFRICA, INDIA, GERMANY

Available continents = NORTH AMERICA, EUROPE, ASIA, AUSTRALIA, SOUTH AMERICA, AFRICA, CENTRAL AMERICA,

As international rank = 20 the university name is = UNIVERSITY OF TORONTO

As national rank = 1 the university name is = UNIVERSITY OF TORONTO

The average score = 75.000

The relative score to the top university in NORTH AMERICA is = $(75.00 / 100.00) \times 1000 = 75.000$

The capital is => OTTAWA

9. The universities that hold the capital name

You need to list all the universities names where the name contains (at any place) the capital name of the selected country. For example, if the selected country is CANADA, then the UNIVERSITY OF OTTAWA should be listed while UNIVERSITY OF TORONTO should NOT be listed. The output should be like the following

The universities that contain the capital name =>

#1UNIVERSITY1

#2UNIVERSITY2

#3 UNIVERSITY3

where UNIVERSITY1, UNIVERSITY2, UNIVERSITY3, are the names of the universities that must contain the complete capital name of the selected country. The following is an example output:

≡ output.txt

Total number of universities in 2000

Available countries = USA, UNITED KINGDOM, JAPAN, FRANCE, GERMANY, SWITZERLAND, SOUTH AFRICA, SPAIN, CANADA

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

AN INTERNATIONAL JOURNAL OF THE UNIVERSITY OF CHICAGO PRESS

As national rate is 3 the property rate is 100 percent of rate

The average score is 76.0%

The interest paid to the tax authority is $2000 \times 10\% = 200$ and $2000 \times 10\% = 200$.

© 2000 Blackwell Science Ltd

The universities that contain the capital name =>

#1 UNIVERSITY OF TOKYO

#2 TOKYO INSTITUTE OF TECHNOLOGY

#3 TOKYO METROPOLITAN UNIVERSITY

tester.py

This file is provided to you so that you will be able to test your work. It basically executes the following steps in the following order:

- Imports the module you completed with the name **univRanking**.
- Calls the one function from your module as follows:
`univRanking.getInformation(selectedCountry, "TopUni.csv", "capitals.csv")`
 - The function name MUST BE **getInformation** and pay attention to the case.
 - The second parameter is the **ranking file name** which is set by default to **"TopUni.csv"**
 - The third parameter is the **capitals file name** which is set by default to **"capitals.csv"**
- The **getInformation** function will extract all the required information according to the specifications explained in the previous section and store this information in the **output.txt** file according to the **exact formatting specified earlier**. Your code will be tested against three countries (USA, South Korea, Japan) and the test result will be displayed for each required information accordingly. After the complete run for **tester.py**, the following will be printed:

```
#####
selectedCountry = 'USA'
The content of output.txt is:
#####

#####
Testing information about USA:
#####
Testing... # 1 - Universities count...successful!
Testing... # 2 - Available countries...successful!
Testing... # 3 - Available continents...successful!
Testing... # 4 - The university with top international rank...successful!
Testing... # 5 - The university with top national rank...successful!
Testing... # 6 - The average score...successful!
Testing... # 7 - The relative score...successful!
Testing... # 8 - The capital city...successful!
Testing... # 9 - The universities that hold the capital name...successful!

#####
selectedCountry = 'South Korea'
The content of output.txt is:
#####

#####
Testing information about South Korea:
#####
Testing... # 1 - Universities count...successful!
Testing... # 2 - Available countries...successful!
Testing... # 3 - Available continents...successful!
Testing... # 4 - The university with top international rank...successful!
Testing... # 5 - The university with top national rank...successful!
Testing... # 6 - The average score...successful!
Testing... # 7 - The relative score...successful!
Testing... # 8 - The capital city...successful!
Testing... # 9 - The universities that hold the capital name...successful!

#####
selectedCountry = 'japan'
The content of output.txt is:
#####

#####
Testing information about Japan:
#####
Testing... # 1 - Universities count...successful!
Testing... # 2 - Available countries...successful!
Testing... # 3 - Available continents...successful!
Testing... # 4 - The university with top international rank...successful!
Testing... # 5 - The university with top national rank...successful!
Testing... # 6 - The average score...successful!
Testing... # 7 - The relative score...successful!
Testing... # 8 - The capital city...successful!
Testing... # 9 - The universities that hold the capital name...successful!
```

Tips and Guidelines

This is a general guideline on how to implement your code.

- First, download all the provided files into your working folder.
- create the file `univRanking.py` in the same working folder. This way you now have 4 files in the same folder.
- Implement the function with the **exact name**:
`getInformation(selectedCountry, rankingFileName, capitalsFileName)`
Start by writing some code to write any text to `output.txt`

- Test if `tester.py` is working and displaying the content of `output.txt` correctly.
- Now, start implementing the required code for the `getInformation` function. This function reads the contents of both `TopUni.csv` and `capitals.csv` and converts it into an appropriate list representation. If any of the files is missing, you must handle this as an exception by printing a "file not found" error message to the output file and exiting the program. You can use the `quit()` function to exit. You can use the following code as a guide to read the files:

```
def loadCSVData(filename):
    # This function is intended to load the content of filename and return the content in a list format. Any exception in this function should be handled.
    list = []
    fileContent = open(filename, "r", encoding='utf8')
    for line in fileContent:
        # if this line is a header (first line) you need to skip it since the header is not part of the data
        # cleanup the line e.g., using strip()
        # split the line into multiple parts e.g., using split(",")
        # select the parts that will be used in the assignment and add them to the list
    fileContent.close()
    return list
```

Don't forget to use `encoding='utf8'` when reading the file to read special characters correctly.

- It is a good idea to have a function that perform some cleanup before using the data by removing the unnecessary columns in the used data files.
- It is also a good idea to combine both files in such a way that you include the capital and continent for each university.
- It will be easier to define at least one function for each of the requirements mentioned above. Then, you call all the functions inside the `getInformation` function.
- It is a very good practice to have some helper functions. Each of these functions is defined to accomplish a relatively small task, while this task may be used by other functions multiple times. For example:
 - A function to return all university info using its country name e.g.,
`info = findUnivByName(selectedCountry)`
 - A function to return the continent name using the country name e.g.,
`continent = findContinentByCountryName(selectedCountry)`
 - And so on..
- Debug...Debug...Debug...
- Constants must be named with all uppercase letters.
- Variables should be named in lowercase for single words and camel case for multiple words, i.e. `extraToppingCost`
- All user input in this program should be case-insensitive, meaning that the user can type in lowercase or uppercase or a mixture, and it should work the same regardless
- Add comments throughout your code to explain what each section of the code is doing and/or how it works
- You can assume the following:
 - All the numbers presented in the examples above are for demonstration purposes only.
 - You must follow the exact labels (wording and case) used in the output requirements. These labels are highlighted in `purple`.
 - All the data that is coming from the .csv files should be displayed as UPPERCASE.
 - The provided examples of the `output.txt` file have some blurred text. This is done for the purpose of showing the corresponding output relative to the other outputs from the other requirements.
 - The selected country is always entered correctly by the user.

Rules

- Read and follow the instructions carefully.
- Only submit the Python file described in the Files section of this document.
- Submit the assignment on time. Late submissions will receive a late penalty of 10% per day (except where late coupons are used).
- Forgetting to submit a finished assignment is not a valid excuse for submitting late.
- Submissions must be done on Gradescope. They will not be accepted by email.
- You may re-submit your code as many times as you would like. Gradescope uses your last submission as the one for grading by default. There are no penalties for re-submitting. However, re-submissions that come in after the due date will be considered late and subject to penalties (or to the use of late coupons).
- Assignments will be run through a similarity checking software to check for code that looks very similar to that of other students. Sharing or copying code in any way is considered plagiarism and may result in a mark of zero (0) on the assignment and/or reported to the Dean's Office. Plagiarism is a serious offence. Work is to be done **individually**.

Submission

Due: Wednesday, November 16, 2022 at 11:55pm

1. The **ONLY ONE** file required from you to submit is **univRanking.py**. **Do not upload any other files.**
2. The submission will be using [Gradescope](#).

There are several tests that will automatically run when you upload your files. Some of the tests are visible to you so it will give you an idea of how well your program is working. However, we will also run some hidden tests so you won't see those tests nor the grade you get from those tests. It is recommended that you create your own test cases to check that the code is working properly for a multitude of different scenarios.

Remember!

- Assignment submission after **11:55 PM** will cause late penalty of 10% per day to be deducted from your mark.
- Submissions through the email will not be accepted at any circumstances.
- Please check back this page whenever an announcement is posted regarding this assignment.
- You can use coupons with this assignment to avoid late penalty.
- If you don't complete the coupon quiz, this means that you are using zero coupons for this assignment.

Marking Guidelines

The assignment will be marked as a combination of your auto-graded tests (both visible and hidden tests) and manual grading of your code logic, comments, formatting, style, etc. Below is a breakdown of the marks for this assignment:

- **[50 marks]** Auto-graded Tests
- **[20 marks]** Code logic and completeness
- **[10 marks]** Comments
- **[10 marks]** Code formatting

- **[10 marks]** Meaningful and properly formatted variables
- **Total: 100 marks**

Remember!

The weight of this assignment is **12%** of the course mark.