Application Development Emerging Technology

Fall 2022

**Final Exam**

**Name: Irida Medina**

**Class: ISMN 6650**

**-------------------------------------------------------------------------------------------------------------------------------**

**Question 1: (15 points)**

Write a program that sorts a list of ***Country*** objects in decreasing order so that the most populous country is at the beginning of the list.

You should submit two Python files: 1: ***Country*** class file; 2: a test file to read ***data.txt***, leverage ***Country*** class, and print out the most populous country.

***Country*** class should store the name of the country, its population, its area, and the population density.

You should leverage ***data.txt*** from folder 1 to work on this question.

In the data.txt, the attributes of each row refer to “country”, “population”, and “area”.

Hint: In your test file, you could use **sorted()** to sort.

**Text

Description automatically generatedCode Question 1:**

**Text

Description automatically generated**

**My sample output – Question 1:**

**Text

Description automatically generated**

**Question 2: (15 points)**

Provide a class ***Letter()*** for authoring a simple letter. In the constructor, supply the names of the sender and the recipient:

***def \_\_init\_\_(self, letterFrom, letterTo)***

Supply a method

***def addLine (self, line)***

to add a line of text to the body of the letter.

Supply a method

***def getText(self)***

that returns the entire text of the letter.

You also need to implement a test file to test your ***Letter().***

**The letter should have the following form:**

Dear *recipient name*:

*“blank line”*

*first line of the body*

*second line of the body*

*….*

*last line of the body*

*“blank line”*

*Sincerely,*

*“blank line”*

*sender name*

For example, assuming that you have the sender named **John** and the recipient named **Mary**. John wrote two lines of the letter: 1: “How are you?” 2: I really miss you!

Your test file leveraging ***Letter***() will have the following output:

**Dear Mary:**

**How are you?**

**I really miss you!**

**Sincerely,**

**John**

**(**Tips: **“\**n” allows you to add a blink line; in addition to have instance variables *“letterFrom”, and “letterTo*”, ensure to have another instance variable that has the empty list to store the line of your input; you may also want to check in class exercise 2 from Lab 9.2 to get the idea of writing your code).

Submit two Python files: 1: ***Letter()***; 2: **a test file to test *Letter()***

**Code Question 2:**

Text

Description automatically generated

Text

Description automatically generated

**My sample output – Question 2:**

Text

Description automatically generated

**Question 3: (15 points)**

Write a program that reads in two files: a ***template.txt*** and a ***database.txt***andwrites junk mails (save your outputs as the **.txt format**) using the template email format. Leverage ***template.txt*** and ***database.txt*** files from folder 3 to work on this.

The template file contains text and tags. The tags have the form |1| |2| |3|… and need to be replaced with the first, second, third, … field in the current database record.

A ***database.txt*** file in this question looks like this:

**Mr.|Harry|Morgan|1105 Torre Ave.|Cupertino|CA|95014**

**Dr.|John|Lee|702 Ninth Street Apt. 4|San Jose|CA|95109**

**Miss|Evelyn|Garcia|1101 S. University Place|Ann Arbor|MI|48105**

A ***template*** file in this question looks like this:

**To:**

**|1| |2| |3|**

**|4|**

**|5|, |6| |7|**

**Dear |1| |3|:**

**You and the |3| family may be the lucky winners of $10,00,000 in the Python**

**clearinghouse sweepstakes. Please visit our offices in |5|, |6| to**

**confirm your winning status.**

**Sincerely,**

**Scammers**

Given that there are three people in the **database** file, your program should be able to generate **three** junk mails automatically. For your output files (i.e., three junk mails), ensure to save the name as “first name + last name”.txt.

In other words, once I execute your codes, there will be three files generated automatically with the file names of ***HarryMorgan.txt*, *HarryMorgan.tx****t*, and ***JohnLee.txt*** in my side.

(Tips: use ***.replace()*** to replace the value and use ***.write()*** to write into the junk mail template; use ***.read()*** to extract a string that contains characters in the file).

**Code Question 3:**

**Text

Description automatically generated**

**My sample output 1 – Question 3 (HarryMorgan.txt):**

**Graphical user interface, text, application, email

Description automatically generated**

**My sample output 2 – Question 3 (JohnLee.txt):**

**Graphical user interface, text, application, email

Description automatically generated**

**My sample output 3 – Question 3 (EvelynGarcia.txt):**

**Graphical user interface, text

Description automatically generated**

**Question 4: (15 points)**

Write a program that reads in ***1.txt* file** from folder 5, coverts all words to lowercase, and prints out all words in the file that contains the letter a, the letter b, the letter c, and so on. Build a **dictionary** whose **keys** are the lowercase letters, and whose **values** are sets of words containing the giving letter.

**Code Question 4:**

**Text

Description automatically generated**

**My sample output – Question 4:**

**Text

Description automatically generated**

**Question 5: (15 points)**

Design and implement a class ***Country*** that stores the name of the country, its population, its area, and the population density (i.e., people per square kilometer (or mile). In other words, there are four major methods in this class (i.e., get country name, get area information, get population information, and get population density information). Then, write a test file that reads ***data.txt*** and prints the following three tasks by leveraging methods from the ***Country()*** class.

* The country with the largest area.
* The country with the largest population.
* The country with the largest population density

You should turn in **TWO** Python files in this question. One is a ***Country*** class file and the other is the test program that leverages the methods from the ***Country()*** class and print the above three requests. You could leverage ***set()*** to store country information and do the operations.

Use dataset from Q4 folder to work on this question.

In the data.txt, the attributes of each row refer to “country”, “population”, and “area”.

**Code Question 5:**

**Text

Description automatically generated**

**Text

Description automatically generated**

**My sample output – Question 5:**

**Text

Description automatically generated**

**Question 6: (20 points)**

Write a program that reads the contents of a text file. The program should create a dictionary in which the key-value pairs are described as follows.

* Key. The keys are the individual words found in the file
* Values. Each value is a list that contains the line numbers in the file where the word (the key) is found.

For example, suppose the word “robot” is found in lines 7, 18, 94, and 138. The dictionary would contain an element in which the key was the string “robot”, and the value was a list containing the numbers 7, 18, 94, and 138.

Once the dictionary is built, the program should create another text file called **index.txt**, known as a word index, listing the contents of the dictionary. This word index file should contain an alphabetical listing of the words that are stored as keys in the dictionary, along with the line numbers where the words appear in the original file.

In other words, once I run your code, one file called **index.txt** should be generated automatically in my folder which contain an alphabetical listing of the words that are stored as keys in the dictionary, along with the line numbers where the words appear in the original file. Use **Kennedy.txt** as the reading input for this question.

**Kennedy.txt:**

Text

Description automatically generated

**Code Question 6:**

Text

Description automatically generated

**My sample output – Question 6:**

**Text

Description automatically generated**

**Question 7: (20 points)**

Write a program that creates a dictionary containing the U.S. states as keys, and their capitals as values. (Use the Internet to get a list of the states and their capitals.)

The program should then randomly quiz the user by displaying the name of a state and asking the user to enter that state’s capital. You could use a **while** loop to let the user continue to guest till a certain condition meets to terminate. The program should keep a count of the number of correct and incorrect responses at the end of round of guess.

You should **import random** first. Then, in your code you need to use **random.randint**() to get a random state name for the question. Also, once you have your capital dictionary created, you could use **iter()** to get access to the list of state names. ﻿

Note you will not get the same outputs like the following examples because you will use random.randint() to get a random state name in each round.

**Code Question 7:**

**Text

Description automatically generated**

**Text

Description automatically generated**

**My sample output – Question 7:**

**Text

Description automatically generated**

**Question 8: (5 points) (single choice) (highlight your answer)**

8.1: (1 point)

Which of the following is true regarding subclasses?

1. A subclass inherits methods from its superclass but not instance variables.
2. A subclass inherits instance variables from its superclass but not methods.
3. **A subclass inherits methods and instance variables from its superclass.**
4. A subclass does not inherit methods or instance variables from its superclass.

8.2: (1 point)

How can you access a value stored in a dictionary?

1. A value can only be accessed using a sequential search
2. **A value can only be accessed using its associated key**
3. A value can only be accessed using its index
4. A value can only be accessed using the in operator

8.3: (1 point)

What is the difference between a list and a string?

1. lists are sequences but strings are not and therefore you cannot access a string element using []
2. lists are immutable, but strings can be changed
3. **lists can hold values of any type, whereas strings are only sequences of characters**
4. elements in a list can be accessed using an integer as the index value, but strings can use any numeric data type for the index

8.4: (1 point)

What is the purpose of using an inheritance hierarchy?

1. **To share common code among the classes**
2. To create objects from concrete classes
3. To create objects from abstract classes
4. To create objects using constructors

8.5: (1 point)

How do you feel confident about Python now?

1. Very confident
2. Confident
3. **Nothing could stop my passion about Python now 😊**