Coursework 1: Pizza, Pizza, Pizza!

DUE ON KEATs by Sunday 6-Nov by 11:55 PM

This is the first coursework assignment for this module.

- It is worth **5% of your final grade** and is scored out of 50 marks.
- There are 4 Parts to this coursework, see at the end of the module for the Point breakdown for each Part.
- Each Part requires you to turn in a Python script (a `.py` file). Be careful to name the Python scripts with the filename given. ***Failure to do this may cause loss of marks!***
- You must do and turn in your own work, there is no collaborating, and no group work allowed on this module.
- Make sure that you properly comment your code, so that the grade can understand what your program is doing. ***Uncommented code will result in loss of marks!***

Read over the Coursework description carefully before starting.

Data Files

On KEATs download the `data.zip` file and uncompress it to a `data/` directory. The `data/` directory should be in the same directory as your Python scripts.

Make sure that your Python scripts read input data files and write output data files to the `data/` directory. Any files written outside of `data/` will be ignored by the grader

Overview: Pizza Shop

This coursework will have you working with XML, CSV, and JSON files with data related to a Pizza Shop! Your scripts will read and write data files, and a set of sample of input and output files are provided for you as an example of the format of the expected output. The scripts will be run from the command line. It may be helpful for you to prototype your code in Jupyter Notebook, but the code you will turn in is a Python script (.py). This is the same process that you followed in the Lab Practicals 2-3.

Part 1: Pizza Menu

Our Pizza Shops keeps the menu options for pizza in an XML data file called `pizza.xml`. You will find an example pizza.xml file in the data.zip provided with this Coursework.

Here is the pizza.xml file:

```
<pizza>
    <shopname>Chipp's Pizza Shop</shopname>
    <sizes>
        <size code="L">Large</size>
        <size code="XL">Extra Large</size>
    </sizes>
    <toppings>
        <topping code="x">Extra Cheese</topping>
        <topping code="m">Mushrooms</topping>
    </toppings>
    <crusts>
        <crust code="thick">Thick Crust</crust>
    </crusts>
</pizza>
```

You can see that pizza.xml provides information:

- the pizza shop name
- the sizes that the pizza comes in

- the toppings that are available for a pizza
- the pizza crusts the pizza can be made of

Write a Python script called `pizza_menu.py` that reads in the `pizza.xml` file as input using the `xml.etree.ElementTree` module you learned about in lecture and in the practicals.

Your script will then ***display*** a menu of pizza options for the pizza shop. You will see the pizza shop name, the pizza sizes, toppings, and crust options are printed.

To see an example of the program output, look at the text file `pizza_menu.txt`:

```
# Chipp's Pizza Shop

## Sizes
- Large
- Extra Large

## Toppings
- Extra Cheese
- Mushrooms

## Crusts
- Thick Crust
```

Be sure to use *exception handling* in case the file is not found.

Part 2: Pizza Report

Write a new Python script called `pizza_report.py` that reads in the `pizza.xml` file and parses the tree structure (again using the `xml.etree.ElementTree module`).

Have your program *calculate* the following from the pizza information:

1. number of **sizes** of pizzas that are available in the pizza shop

2. number of **toppings** that are available in the pizza shop

3. number of **crusts** that are available in the pizza shop

4. total number of *different types* of pizzas are possible with the following menu. Assume the following about each pizza:

   - a pizza can only be 1 size (i.e. no Large and Extra Large Piza)
   - a pizza can have any combination of toppings, including *all toppings* and including *no toppings*
   - toppings will be on the whole pizza (i.e. no half pepperoni, and half extra cheese pizza)
   - a pizza can only be 1 type of crust

Using the python `csv` module, output your calculations as a single row of data to a CSV file with the name `pizza_report.csv`. Assume the following header for your CSV file along with the calculated numbers from the previous example:

```
sizes,toppings,crusts,total_combo
2,2,1,4
```

The above example is also available in the data.zip named `pizza_report.csv`. Like in the previous example, be sure to use *exception handling* in case the file is not found.

Part 3: Pizza Specials

The pizza shop also stores Pizza Specials in a file `pizza_specials.csv, with the following format:`

```
name,size,toppings,crust
Supreme,XL,xm,thick
Simple Cheese,L,x,thick
```

Every pizza special has a name and designates a **size**, a combination of **toppings**, and a **crust**. The size, toppings, and crust are referenced by a

`code` *attribute*, which is specified in the `pizza.xml` XML file. For example, the topping code for Mushrooms is `"m"`, and the topping code for Extra Cheese is `"x"`. The Supreme pizza has both Mushrooms and Extra Cheese as designated by the string `"xm"`.

Write a new Python script called `pizza_specials.py` that reads in the `pizza.xml` file (again using the `xml.etree.ElementTree module`) and stores the sizes, toppings, and crusts in dictionaries with the code for that element as the *key* and the text for the element as the *value*.

Using these dictionaries, convert the Pizza Specials in the `pizza_specials.csv` file to an menu text description. Write each special's menu description as a line in a text file called `pizza_specials.txt`:

```
Supreme: Extra Large Pizza with Extra Cheese and Mushrooms and Thick Crust
Simple Cheese: Large Pizza with Extra Cheese and Thick Crust
```

See the example file `pizza_specials.txt` in the `data.zip`.

Part 4: Pizza Decider

The final part of the coursework will use a data set collected from the Reddit group "*Random Acts of Pizza*" (https://www.reddit.com/r/Random_Acts_Of_Pizza/). Random Acts of Pizza is a community on the website Reddit.com that facilitates the sending and receiving of pizzas between strangers. People write a request for a pizza on the Reddit group and someone may accept their request and order them a pizza!

A data set has been collated for the textual requests to this Reddit Group, and a simplified excerpt for 10 requests to "*Random Acts of Pizza*" has been provided in the text file `random_acts_pizza.csv`. There are 3 columns in this CSV file:

- `username` - the name of the user requesting pizza
- `text` - the text of the pizza request written by the user
- `received_pizza` - a Boolean whether or not the pizza request was accepted

In this Part you will write a simple Python class that will use this the `random_acts_pizza.csv` file to automate to *decide* whether a new request for pizza should be accepted or not.

1. Create a new python script called `pizza_decider.py`

2. Using the `csv` python module read in the "Random Acts of Pizza" request history that is contained in the *random_acts_pizza.csv* file.

3. Add a class called `PizzaDecider` to your program, which using the request history read in the previous step will decide whether or not new requests for pizza according to two criteria:

   i. The user requesting a new pizza (identified by their username) have not previously received a pizza from the "*Random Acts of Pizza*" group.

   ii. The text of the user's pizza request is longer than 400 characters in length.

4. Using the `json` python module, have your program read in a file called `pizza_request.json`, which contains one request for pizza. The request file is a dictionary which has two keys:

   `username` - the name of the user requesting the pizza

   `text` - the text of the pizza request by the user

   For an example pizza request see the file, `pizza_request.json` in the `data.zip`.

5. Your program will pass the pizza request read in from the `pizza_request.json` file, to a method in the `PizzaDecider` class that will return a boolean whether or not the user should get a pizza according to the criteria above.

6. Finally, your program will write out the decision from the `PizzaDecider` to a new JSON file called `pizza_decision.json` which which is a dictionary with the following keys:

`username` - the name of the user requesting the pizza

`text` - the text of the pizza request by the user

`received_pizza` - a Boolean whether or not the pizza request was accepted

See the file `pizza_decision.json` in the data.zip for the pizza decision from the `pizza_request.json` example mentioned earlier.

What to turn in

Each Part has a python script which you will turn in. Include a header comment on each python script that include the following information:

- Your Name
- Your Student Number
- A short description of what this script does

Part 1	`pizza_menu.py`
Part 2	`pizza_report.py`
Part 3	`pizza_specials.py`
Part 4	`pizza_decider.py`

Put all of these python scripts into a .ZIP file and upload this file on KEATs in the Coursework 1 Submission Box.

Point Breakdown

There are **50 marks** available for this piece of coursework. Each Part is scored accordingly:

Part 1	15 Marks
Part 2	15 Marks
Part 3	10 Marks
Part 4	10 Marks

A **rubric** containing guidelines for programming marks will be distributed separately and also available on KEATS.

COLLEGE POLICY ON LATE SUBMISSIONS

Submissions uploaded *within 24 hours* of the deadline will be marked no higher than **40%**.

Submissions uploaded *after 24 hours* of the deadline will be marked 0, unless mitigating circumstances occur. A mitigating circumstances form must be filed in the Informatics department office, room S5.01.

See http://www.kcl.ac.uk/mitcircs for more details.