Assignment #1
CSCI 201 Spring 2020
4.0% of Course Grade

Title
Restaurant Searching

Topics Covered

Java Classes	Basic Java Topics
File I/O	JSON Parsing
Sorting	

Introduction
Much of your computer programming experience is likely using C++ (at least if you took CSCI 103 and CSCI 104). In this course, Java is the language we will primarily use. To help transition your knowledge from C++ to Java, you will be creating a program that looks up restaurants and menu items based on user search criteria. This assignment requires you to parse a file containing a set of restaurants and their respective menu items. To ensure accurate parsing, you will then provide a command line interface to allow a user to query the parsed data.

Data Persistence
Many software projects need to store persistent data, and there are a few ways to do this. Databases, which we will cover later in the class, are used for many applications, but storing data in files is also common. This assignment will require you to parse a file that contains restaurants and various information about them. The data in the file is going to be stored as a ***JavaScript Object Notation (JSON)*** file, formatted as follows:

```
{
  "data": [
    {
      "name": "Northern Cafe",
      "address": "2904 S Figueroa Street",
      "latitude": 34.025550,
      "longitude": -118.277440,
      "menu": [
          "orange chicken",
          "rice",
          "noodles",
          "dumplings"
        ]
    }
  ]
}
```

There could be multiple items in the *data* array, but there will be at least one. The same is true for the *menu* array; there will always be at least one item on the menu. In this example, there is only one object in the *data* array, and four items in the *menu* array. A sample file is provided in Blackboard. I recommend that you create your own test files that are longer since the file we will use for testing will be different from the sample one provided.

Parsing JSON

JSON is a lightweight data-interchange format. In other words, it is a syntax that allows for easy storage and organization of data. It is commonly used to exchange information between client and server, and it is popular because of its language independence and human readability. JSON is built upon two basic data structures that you should already be familiar with: dictionaries (maps) and ordered lists. An object in JSON is represented by an unordered set of name/value pairs (i.e. dictionary). Objects are contained by braces `{ }`, inside of which will list the object's attributes (with the syntax `name : value`), using a comma as the separator.

There are quite a few Java JSON parsing APIs out there. Unfortunately, the JDK does not have built-in support for JSON, but some of the more popular ones include GSON, Jackson, and JSON.simple. Here are a couple of blogs discussing the merits of choosing one of these APIs over another:

http://blog.takipi.com/the-ultimate-json-library-json-simple-vs-gson-vs-jackson-vs-json/
http://javarevisited.blogspot.com/2016/09/top-5-json-library-in-java-JEE.html

In short, you may want to start with GSON because it is known for its ease and flexibility of converting Java objects into JSON objects (and vice versa). No matter which API you choose (and you are not limited to choosing from the ones mentioned in these instructions), you will need to download a JAR file and add it to your Eclipse project. Below are links to the JAR file downloads for the APIs mentioned above:

GSON:
https://github.com/google/gson

Jackson:
https://github.com/FasterXML/jackson-core

JSON.simple:
https://github.com/fangyidong/json-simple

You will need to add the JAR file to your Java Build Path in Eclipse to use the library. First, move the JAR file to the top directory of your Eclipse project. In other words, if your project is named "Assignment1", put the JAR file into the "Assignment1" directory in your Eclipse workspace directory. Next perform the following steps in Eclipse:

1. Right click on your Eclipse project.
2. Click "Refresh", which should make your JAR file show in the Package Explorer.
3. Right click on your Eclipse project again.

4. Choose “Properties”.
5. Select “Java Build Path”.
6. Click the “Libraries” tab.
7. Click “Add JARs”.
8. Find the JAR in your project directory and add it.
9. Click “Okay”.

Calculating Distance

Geolocational coordinates will be provided for each restaurant. The JSON file will store the latitude and longitude for each location, and you will need to calculate the distance (in miles) whenever the distance is displayed. Here’s a formula to calculate the distance between two latitude/longitude coordinates (in miles):

```
d = 3963.0 * arccos[(sin(lat1) * sin(lat2)) + cos(lat1) * cos(lat2)
* cos(long2 – long1)]
```

Assignment

When your program first runs, you will need to prompt the user to enter the name of the JSON file that contains the data for the restaurants. Your program should validate that the file is formatted properly. You can assume the following data types for each:

```
data : Array (of JSON Objects)
name : String
address : String
latitude: double
longitude: double
menu : Array (of Strings)
```

If there is any problem parsing the data in the file, your program should print out as descriptive of an error message as possible. The program should then prompt the user to enter another file. Here are some examples of errors in file parsing you need to catch:

- File not found
- Data cannot be converted to the proper type as shown above
- Missing data parameters

Once a properly-formatted file is parsed, your program should ask the user for his/her current coordinates. Your program should then display a menu as follows:

```
            1) Display all restaurants
            2) Search for a restaurant
            3) Search for a menu item
            4) Add a new restaurant
            5) Remove a restaurant
            6) Sort restaurants
            7) Exit
      What would you like to do?
```

Users can enter menu commands by typing the number associated with the command, i.e. entering "1" to display all the restaurants. For all the user input in this program, ***the data entered should be case-insensitive, except for the file names.*** Partial matches should be accepted, and multiple results can be returned from options 1, 2, 3, and 6.

Sample Execution

Here is a sample execution of the program with the user input bolded (though the input will not be bolded when you run your program). Assume restaurant.txt contains the data from the sample file provided.

```
What is the name of the restaurant file? badfile.txt
The file badfile.txt could not be found.

What is the name of the restaurant file? badformat.txt
The file badformat.txt is not formatted properly.

What is the name of the restaurant file? restaurant.txt
The file has been properly read.

What is your latitude? 34.021160

What is your longitude? -118.287132

            1) Display all restaurants
            2) Search for a restaurant
            3) Search for a menu item
            4) Add a new restaurant
            5) Remove a restaurant
            6) Sort restaurants
            7) Exit
What would you like to do? 0

That is not a valid option.

What would you like to do? hello
```

```
That is not a valid option.

What would you like to do? 1

Northern Cafe, located 0.8 miles away at 2904 S Figueroa Street
The Lab Gastropub, located 0.4 miles away at 3500 S Figueroa Street
Taco Bell, located 0.5 miles away at 3629 S Vermont Ave

           1) Display all restaurants
           2) Search for a restaurant
           3) Search for a menu item
           4) Add a new restaurant
           5) Remove a restaurant
           6) Sort restaurants
           7) Exit
What would you like to do? 2

What is the name of the restaurant you would like to search for? Cup
of Joy

Cup of Joy could not be found.

What is the name of the restaurant you would like to search for?
northern cafe

Northern Cafe, located 0.8 miles away at 2904 S Figueroa Street

           1) Display all restaurants
           2) Search for a restaurant
           3) Search for a menu item
           4) Add a new restaurant
           5) Remove a restaurant
           6) Sort restaurants
           7) Exit
What would you like to do? 3

What menu item would you like to search for? waffles

No restaurant nearby serves waffles.

What menu item would you like to search for? chicken

Northern Cafe serves orange chicken.
The Lab Gastropub serves fried chicken and chicken sandwich.
Taco Bell serves chicken taco.

           1) Display all restaurants
```

```
           2) Search for a restaurant
           3) Search for a menu item
           4) Add a new restaurant
           5) Remove a restaurant
           6) Sort restaurants
           7) Exit
What would you like to do? 4

What is the name of the restaurant you would like to add? Northern
Cafe

There is already an entry for Northern Cafe.

What is the name of the restaurant you would like to add? Cup of Joy

What is the address for Cup of Joy? 3016 S Figueroa Street

What is the latitude for Cup of Joy? 34.024899

What is the longitude for Cup of Joy? -118.277977

What does Cup of Joy serve? thai tea

     1) Yes
     2) No
Does Cup of Joy serve anything else? 1

What does Cup of Joy serve? oolong milk tea

     1) Yes
     2) No
Does Cup of Joy serve anything else? 1

What does Cup of Joy serve? chicken strips

     1) Yes
     2) No
Does Cup of Joy serve anything else? 2

There is now a new entry for:
Cup of Joy, located 0.7 miles away at 3016 S Figueroa Street
Cup of Joy serves thai tea, oolong milk tea, and chicken strips.

           1) Display all restaurants
           2) Search for a restaurant
           3) Search for a menu item
           4) Add a new restaurant
           5) Remove a restaurant
```

```
           6) Sort restaurants
           7) Exit
What would you like to do? 5

     1) Northern Cafe
     2) The Lab Gastropub
     3) Taco Bell
     4) Cup of Joy
Which restaurant would you like to remove? 2

The Lab Gastropub is now removed.

           1) Display all restaurants
           2) Search for a restaurant
           3) Search for a menu item
           4) Add a new restaurant
           5) Remove a restaurant
           6) Sort restaurants
           7) Exit
What would you like to do? 6

     1) A to Z
     2) Z to A
     3) Closest to farthest
     4) Farthest to closest
How would you like to sort by? 3

Your restaurants are now sorted from closest to farthest.

           1) Display all restaurants
           2) Search for a restaurant
           3) Search for a menu item
           4) Add a new restaurant
           5) Remove a restaurant
           6) Sort restaurants
           7) Exit
What would you like to do? 1

Taco Bell, located 0.5 miles away at 3629 S Vermont Ave
Cup of Joy, located 0.7 miles away at 3016 S Figueroa Street
Northern Cafe, located 0.8 miles away at 2904 S Figueroa Street

           1) Display all restaurants
           2) Search for a restaurant
           3) Search for a menu item
           4) Add a new restaurant
           5) Remove a restaurant
           6) Sort restaurants
```

```
           7) Exit
What would you like to do? 7

     1) Yes
     2) No
Would you like to save your edits? 1

Your edits have been saved to restaurant.txt.
Thank you for using my program!
```

Grading Criteria

The manner by which you go about implementing the solution is not specifically graded, but the output must match exactly what you see in the execution above.

File I/O (0.8%)

0.2% - The filename is read from the user, and the file is appropriately parsed
0.1% - If the file could not be found, an appropriate error message is displayed
0.1% - If the file cannot be parsed, an appropriate error message is displayed
0.1% - If the file has missing parameters, an appropriate error message is displayed
0.3% - The file is properly saved upon exiting the program

Reading Inputs (1.2%)

0.2% - The program prompts the user for their latitude and longitude coordinates
0.3% - Invalid user input is handled properly
0.7% - Users can properly navigate the menu options by using numerical inputs

Outputs (1.5%)

0.2% - “Display all restaurants” displays all restaurants from the file
0.1% - “Search for a restaurant” displays the correct information for the search query
0.1% - “Search for a menu item” displays the correct information for the search query
0.3% - “Add a restaurant” works properly
0.1% - “Remove a restaurant” works properly
0.4% - Each of the four methods for sorting restaurants works properly
0.3% - Partial searching for restaurant and menu items is supported properly

Calculating Distance (0.5%)

0.3% - Distance is correctly calculated using the latitude and longitude coordinates
0.2% - Only latitude and longitude coordinates are read from and saved to the input file