CST 8130 Data Structures

Assignment #3

Purpose Demonstrate use of:

- Use of good class construction including packages, javadoc & comments and other industry best practices
- Use of file I/O with secondary storage along with the *Java Collections Framework*

Task Read a .csv file, store the data in a Map, then retrieve elements from the Map

---

Detailed requirements:

- You are provided with a .csv file containing data on postal codes for Ottawa and surrounding towns/neighbourhoods in Ontario.
  - Each line of the file corresponds to one postal code.
  - The fields/columns are: *prefix*, *place-name*, *province*, *latitude* & *longitude*.
- And you provided with a main() class which opens the .csv file, reads each line and displays it on the "console". (You can change *this* file).
- Lastly you are provided with a class which encapsulates the data from one line of the file and a method that can randomly generate a postal code prefix
  - PostalCodeEntry.java
    - Do **not** change this file !
- Your task is to:
  - Input the .csv file
  - Store each instance of PostalCodeEntry.java in a concrete implementation of Map
    - Use the prefix as the "key".
    - Choose an appropriate implementation that focuses on searching.

    - Test the Map
        - Use the static method PostalCodeEntry.getRandomPrefix()
        - Create a loop which generates 10 random prefixes.
        - Retrieve the PostalCodeEntry from the Map that is associated with each random prefix.
- Your output must look like the ***Sample Output*** (screenshot provided on Brightspace).
    - The actual values may differ, but your output must look like the ***Sample Output***
    - Hint: use System.out.printf() for all output

Coding requirements:

- follow the Java Coding Conventions as your "style guide"
- create at least 2 classes
    - name of the class with the main() method must end in "Test"
- create at least 2 packages
- the main() method should perform all I/O
    - there is no end-user input, there should only be output
- export your Eclipse project as a .zip
    - name your file using the convention: Lastname.Firstname.Assignement3.zip
    - example: Flinstone.Fred.Assignment3.zip
- Similarly, be sure to name your project following the convention: Lastname.Firstname.Assignment3
- and submit your solution via Brightspace

Note:

- You will lose marks if you do not:
    - follow the instructions under ***Detailed Requirements***
        - including any API features or algorithms described
    - meet the ***Coding requirements***
    - ensure there are complete Javadoc comments for all public constructors and methods.
    - produce the same output as the ***Sample output***