Bryn Mawr College
CMSC 113 Computer Science 1
Spring 2025
Assignment#6
Due on Monday, April 28, 2025

Title: Searching US Postal Zip Code Locations & Doing Object-oriented Design

Purpose: Write a complete working Java program to repeatedly input a zip code from the user. In response, your program will output the name of the town and the state to which it belongs. In case the zip code doesn't exist, it tells the user about it (see example interaction below). The dataset of all the zip codes in the United States is available in the file:

Data File: ~dkumar/CMSC113/Assignments/Assignment6/uszipcodes.csv
Test Data File: ~dkumar/CMSC113/Assignments/Assignment6/testZip.csv

Description: The data file provided (**`uszipcodes.csv`**) contains over 40,000 US Postal Zip Codes and the names of the town and state they belong to. Initially, while you are developing and testing your program, you should use the much smaller (only 15 zip codes) data file (**`testZip.csv`**). Here are the details of the data file's format:

The first line of the data file is a bunch of comma-separated values in the format shown below:

```
NUM_ZIPCODES,zip,city,state
```

These indicate the following:

`NUM_ZIPCODES` is the total number of zip codes present followed by the text field names: **`zip,city,state`**

The rest of the data file contains entries (one line for each zip code) in the following format:

```
ZIP_CODE,TOWN_NAME,STATE
```

`ZIP_CODE` is the 5 digit zip code
`TOWN_NAME` is the name of the town
`STATE` is the name of the town and the 2-letter state abbreviation.

Here are some example entries:

```
56315,Brandon,MN
56316,Brooten,MN
56317,Buckman,MN
56318,Burtrum,MN
56319,Carlos,MN
```

```
56320,Cold Spring,MN
```

The first line above is for the zip code **56315** which belongs to the town of **Branden**, **MN**.

What to do:

Create a class called **Place** to model each zip code to contain the following instance variables:

```
zipCode, town, state
```

Additionally, you will need to define the needed **accessor**, and **modifier** methods. Also, define the print method **toString()** and a comparison method **isPlace()** to compare a given zip code against a place object. Here is the design of the **Place** API (type/class/object):

```
Place - class

Place (String z, String t, String s)
- Constructor. Creates an object with the given zip z, town t, and state s

String getZip()- Accessor. Returns the zip code of this place object
String getTown() - Accessor. Returns name of the town for this object
String getState() - Accessor. Returns name of state (2-letter abbrev)

boolean isPlace(String zip) - Does this object have the given zip?

String toString() - Print Method. Returns a string of form "Bryn Mawr, PA 19010"

main() - A test program (see below)
```

The test program should read in the test data file (with 15 entries) and print out all the towns in the format "**Bryn Mawr, PA 19010**".

Once you have designed, implemented, and tested the above. Write a client program (in a separate file called **SearchZips.java**, that:

- Defines an array of place objects (**Place[] places**)
- Uses a function **readData(Place[] p, In infile)** to read the data file
- and fills up the array, **p[]** with place objects.

NOTE: You will need to pass an extra parameter of type **In** (see description of **readData()** above) to denote the file the data is being read from.

Next, your program should interact with the user as shown below (items in bold are user inputs):

```
$ java-introcs SearchZips
"/home/dkumar/CMSC113/Assignments/Assignment7/uszipcodes.csv"

Read 42613 items from file:
~dkumar/CMSC113/Assignments/AAssignment6/uszipcodes.csv

Search for zip code: 19010
The zip code 19010 belongs to Bryn Mawr, PA
Do you want me to search again? Yes

Search for zip code: 99400
The zip code 99400 does not exist.
Do you want me to search again? Yes

Search for zip code: 91729
The zip code 91729 belongs to Rancho Cucamonga, CA
Do you want me to search again? No

Good Bye!
```

Hints

- Use the design methodology learned in class (and described in Chapter 3 of your text)
- Design and implement your program incrementally, testing each time. DO NOT try to write and test the entire program all at once.
- See page 354 of your text for a description of useful File Input API.
- You will need to use the String method **`split()`** to split the input (each line is a list of comma-separated values (csv)).
- You are required to use the OOP design as specified above for the **`Place`** class.
- You do not need to copy or edit the input data file provided.

What to Hand-in:

1. Once done, hand in a printout of your program code along with a printout showing an interaction for searching the following zip codes: 19010, 16046, 74446, 90213, 68940, 87901, 86501, 95975, 36254, 97009.
2. Write a short reflection on your experiences with this assignment.
3. Staple all these together and hand them in at the start if the class on the due date.