

In this lab, you will practice with methods.

1. Create a project named Lab7 with a class named Lab7Main. Copy the file "paPop.csv" from Canvas into your main project directory. This file is a comma-separated value (CSV) file that contains information about the population density of Pennsylvania over a range of years. The format is <year,density>, one set per line, where year is an integer and density is a double.2. Create a class named YearPop that contains two fields: int year and double pop. It should have an overloaded constructor that sets the member data, and it should have getters for the data (but not setters).

3. Lab7Main should have this member data:

- a File object named myfile, set to null;
- a Scanner object named fileScanner, set to null
- an ArrayList<YearPop> named data to hold the file data, set to null

Create these methods in Lab7Main (Note: except for main(), these are ***not*** static methods. Use regular methods.):

- boolean openFile(filename) that takes a String parameter called filename and new's up myfile using filename. Then, in a try-catch block, as we've done in other code, new up fileScanner with myfile. If that fails, display an error message and return false (instead of calling exit() like we did before), otherwise return true.
- YearPop makeData(line) that takes one String parameter named line in the CSV format above, splits it, creates a new YearPop object with that data in it, and returns that object.
- void createList() should new up data; use a loop to read lines from the data file, one at a time; call makeData on each line; then place the returned YearPop object into the data ArrayList.
- double findYear(year) should look for year in the ArrayList data; if it finds it, return the population density; if not, return -1.0 (an error code).
- main(): new-up a Scanner object for reading from the keyboard; new-up a Lab7Main object called lab and call openFile() to open the data file; call createList() to read the data; prompt the user for a year; use findYear() to get the population density and display it like this:
Year: 1908
Population density: 165.073

then ask the user if they want to look up another (use Y or N) – loop until the user enters N.

4. This has nothing to do with the rest of the lab; it's just recursion practice. Write a **static** recursive method with this signature:

```
public static int computeFibonacci(int first, int second, int n)
```

that computes the n'th entry of the Fibonacci sequence, which is (normally) 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ... This is generated by starting with two integers (normally 0 and 1), adding them together to get the next value (1), adding those next two to get the next value (2), and so on. So if the program calls

```
computeFibonacci(0,1,7)
```

it should return 13 (counting from 0, of course), and you should display:

```
Fibonacci #7 = 13
```

The call could set first and second to anything, but always call it with 0 and 1 to keep things simple; try it out with different values of n. The stopping condition is: when the parameter n is 0, just return first, but for some larger value of n, compute the n-1'th Fibonacci number by recursion.

Note: this is a pretty common problem, so you *could* find the solution on the web, but don't – try to code this to get some practice with recursion. You might want to print the values of the parameters for debugging purposes.

Deliverable: Add your name and Andrew id to the comment at the top of the LabMain.java file. Upload the file to Canvas (no need to upload YearPop.java).