

Specifications

Problem Description

You are going to write a program that reads U.S. state education data from a file into an array of objects of class `StateEducationData`, provided for you. You will then display statistics of the data, including averages, the state with the highest high school completion rate, etc.

Input:

The file `State Education Data.txt` will contain data for exactly ten states. You may not assume what state data is in the file, but you may assume that the file has the format described below.

The file contains data for exactly ten states, ordered as name, high school completion percentage, college completion percentage, and advanced degree completion percentage. For example, eight lines of the file might look like:

```
Arizona
86.2
28.0
10.4
Arkansas
85.2
21.5
7.6
```

You do not need to validate the data in the file. You may assume ten complete records of the correct data types.

(Data is from the US Census Bureau's American Fact Finder:

<https://factfinder.census.gov/faces/nav/jsf/pages/index.xhtml>)

The appendix of this assignment includes a sample of ten state data records. These may or may not be used to test your program.

Output:

The program will display a list of all of the states in the file with the percentage of individuals who completed high school, the percentage who completed a bachelor's degree, and the percentage who completed an advanced degree. It will list the state with the largest high school completion percentage, the lowest high school completion percentage, and it will display the average high school completion percentage.

Interface:

The program will display the output according to this example:

Name :

Alabama

Percent Completed HS:	84.8
Percent Completed Bachelor's:	24.0
Percent Completed Adv. Degree	8.9

Name:	Alaska
Percent Completed HS:	92.3
Percent Completed Bachelor's:	28.8
Percent Completed Adv. Degree	10.4

... (etc., for all ten states)

The state with the highest high school completion percent is:

Name:	Alaska
Percent Completed HS:	92.3
Percent Completed Bachelor's:	28.8
Percent Completed Adv. Degree	10.4

The state with the lowest high school completion percent is:

Name:	California
Percent Completed HS:	82.1
Percent Completed Bachelor's:	32.0
Percent Completed Adv. Degree	11.9

The average high school completion percent for all states is:

87.4

Design specifications:

Use only techniques that are covered in the first six lessons. Use file input and console output.

Do not, under any circumstances, use `System.exit` or more than one `return` per method. Use `break` only in `switch` statements. Do not submit code that includes these constructs (except for `break` in `switch` statements). Submit whatever you can write that does not include these constructs.

Meeting specifications and making good design choices are critically important.

StateEducationData.java

This is a class diagram of the `StateEducationData` class. You must use this class in your program to store the data for each state read from the file. This class is provided for you as a `.class` file in the class website. You should not implement this class yourself. Use the class that's provided. When I test your program, I will test it with the `.class` file that I provided. (Reminder: a `.class` file is a compiled `.java` file. You won't be able to view it with a text editor.) Write a testing program and test

all of these methods, so you understand how they work before you use this class in your program.

StateEducationData
<ul style="list-style-type: none">- name : String- highSchoolPercent : double- collegePercent : double- advancedPercent : double
<ul style="list-style-type: none">+ StateEducationData(stateName : String; highSchoolPercent : double; collegePercent : double; advancedPercent : double)+ getName() : String+ getHighSchoolPercent() : double+ getCollegePercent() : double+ getAdvancedPercent() : double+ setHighSchoolPercent(double newPercent) : void+ setCollegePercent(double newPercent) : void+ setAdvancedPercent(double newPercent) : void+ toString() :String

StateEducationClient.java

readFile method

This program will start by invoking a method that will read the contents of the `State Education Data.txt` file into an array of ten `StateEducationData` objects. The method will return the array of objects. You may not assume what state data is in the file, but you may assume that the file is correctly formatted with data of the correct type and includes ten complete records.

showAll, showHighHSPercent, showLowHSPercent, showAverageHSPercent methods

The program will then invoke each of these methods in order. The program must pass the array of `StateEducationData` objects as a parameter to each of the methods. The methods will display results directly to the console.

Hints

Begin by hard-coding some `StateEducationData` objects and testing the methods. They match the class diagram. Make sure you understand how they work and how to use them.

Next, create an array of `StateEducationData` objects. Write the algorithm to print all of the objects in the array first, so that you have some output you can look at. You can use the `StateEducationData toString()` method to display the information shown for each state.

Next, put the program structure in. Write the method to read the data from a file. Even if you're still hard-coding the data in at this point, invoke the method and return the array. Put the functionality to display all states into a method. Pass the array in to that method.

Once you have that structure in place, start fleshing it out by writing the other methods and testing them, one at a time. Remember your loop variable roles when writing the average, max, and min methods. You will use the accumulator and most wanted holder patterns.

Testing

Test Case 1

Purpose

Test on an average-case file, such as the data in the appendix.

Input

The data in the appendix.

Expected output

See the interface section, above.

Test Case 2

Purpose

The state with the highest high school graduation is first in the file.

Input

Same data, but Alaska first.

Expected output

The same as the interface section, except Alaska's data will be listed first. Max, min, and average should not change.

Rubric

An exceptional-quality assignment will meet the following standards:

- Meeting functional and design specifications, 70 points
The Java program works and meets all of the specifications, with no additional unspecified functionality. The programmer has used programming techniques from the first through sixth lessons only. The required methods are written and used to specifications. The class file is used appropriately. If the program misses specifications or does not function correctly, errors are acknowledged with a thorough and reflective analysis in the testing section (points will be removed for missed specifications).
- Communicating with identifiers and white space, 10 points
The program makes appropriate use of variables. Variables, methods, and constants are named according to convention and are named for understandability and purpose. White space, both vertical and horizontal, is correctly used for readability and meets programming conventions.
- Communicating through documentation, 10 points
The Java program contains comments including the programmer's name and date. There are

block comments (as many as necessary) for each distinct block of code which accurately describe what the block is accomplishing by relating the code to the problem being solved. Javadoc is included at the top of the program and for each method and meets the javadoc standards.

- Assumptions and Testing, 10 points

Testing is thorough. If there are errors, they are described in the testing section. If there are questions, they are answered thoughtfully in the testing section. All assumptions are made explicit.

Copyright © 2020 Margaret Stone Burke and James Burke, All Rights Reserved.

Appendix: Sample state data

Alabama

84.8

24.0

8.9

Alaska

92.3

28.8

10.4

Arizona

86.2

28.0

10.4

Arkansas

85.2

21.5

7.6

California

82.1

32.0

11.9

Colorado

91.0

38.7

14.3

Connecticut

90.1

38.0

16.8

Delaware

88.8

30.5

12.4

Florida

87.2

27.9
10.0
Georgia
85.8
29.4
11.1

Copyrighted Material