CSCI-101 Programming I
Practice Exam 3

Instructions

Please follow the rules below as you work through this exam.

- Please leave all notebooks and electronics (including cell phones and smart watches) at the side of the room.
- This is a closed book/closed notes exam.
- Do not spend too much time on any one problem. You have 50 minutes to complete this exam.
- Partial credit is awarded.
- Please write legibly. If I cannot read your answers, I cannot give you credit.
- Please write your answers in the order specified. If you need additional paper, please raise your hand to ask your instructor for additional paper.
- Your code must be written to behave as specified.
- You must properly use all identifiers that are explicitly stated.
- Please use proper and consistent coding conventions (indentation, naming identifiers, etc.).
- Please stay in your seat until you are ready to hand in your exam. You may leave when you are finished.
- Once you leave the testing room you cannot return until the exam is over. If you need to use the restroom, please use it now.

Point3D.java
Write a class named **Point3D** that models a 3-dimensional point in space. The class should contain the following.

1. Three private fields that hold decimal values for the x, y, and z coordinates of a 3D point.
2. A constructor that takes three decimal values as arguments and initializes the corresponding fields.
3. Getters and setters for each of the fields.
4. A method named **toString** that overrides the **Object** class' **toString** method. The method should return a string containing the three coordinates.
5. A method named **equals** that overrides the **Object** class' **equals** method. Two instances of the **Point3D** class are considered equal if their respective coordinates are equal.

data.txt
Suppose a file named **data.txt** holds contains coordinates for an arbitrary number of points. Each point is written on a separate line. The three coordinates for a point a written with commas between them.

Sample Input File

```
1.0,1.0,1.0
1.0,3.5,1.0
7.3,3.5,1.0
7.3,1.0,1.0
```

Note: the above sample input file is a sample. **Data.txt** can contain any arbitrary number of points, not necessarily 4 as shown above.

App.java
Write a class named **App** that satisfies the following.

1. The class contains a method named **printPoints** that has 2 parameters. The first parameter named **array** holds a reference to an array of **Point3D** objects. The second parameter named **k** holds an integer. The method should use the **Point3D** class' **toString** method to print to the screen the coordinates of the first **k** points in the array.

2. Write a method named **main** that does the following:
   - Allocate an array that can hold 10 **Point3D** references.
   - Declare an integer named **count** that keeps track of the number of **Point3D** references that have been added to the array.
   - Create a Scanner that can read the data in **data.txt**.
   - Read the data in **data.txt** and store the data as **Point3D** objects in the array.
   - Call **printPoints** to print the coordinates of all of the points that are stored in the array.