Name: _____ Lab Section: 06

CS161 Fall 2015 Programming Exam

Write your name above and return this paper to your lab instructor when you submit your exam. You must finish and submit your work using Check-In by the end of this lab.

Do your own work. Only look at your own computer screen. Do not browse, email, or use cell phones and other digital technology other than the computers in the lab.

During the exam you may use your Java text book (or ebook version if you let your instructor know), the Java API documentation website and the course website, but no other web pages.

Your grade will be based on:

- Correctness of each method
- Clarity of code and comments. Your code should be easy to read, and if there are non-obvious steps, they should be explained with comments.
- Object-oriented design: Does your code capture the idea of encapsulation
- Testing: Does your main method adequately test all aspects of your code?

You will write four java classes for keeping track of multiple stars and galaxies in the sky. The names of the classes are Sky, AstroThing, Star, and Galaxy. The classes Star and Galaxy must extend the class AstroThing, to allow an instance of Sky to use a single ArrayList to hold instances of Star and Galaxy.

Your classes must implement the following methods.

Sky: class that is the primary application that you will run.

[**3 points**] **constructor:** Creates an empty ArrayList for holding items of type AstroThing.

[**3 points**] **add:** Add a new AstroThing, either a Star or a Galaxy.

[**3 points**] **toString:** Returns a string representing all stars and galaxies that have been added to the sky.

[**2 points**] **main:** A main method that tests each method in this class.

[**4 points**] **AstroThing:** parent class of Star and Galaxy.

[**4 points**] **constructor:** Has two arguments, a String and a double, for the name and distance from earth for the new AstroThing. Assigns these values to protected instance variables.

[**4 points**] **getName:** Returns name.

[**4 points**] **setName:** Changes the value of name.

[**4 points**] **getDistance:** Returns the distance from earth.

[**4 points**] **setDistance:** Changes the value of distance from earth.

[**4 points**] **equals:** Returns true if two objects being compared have the same name. This must correctly override the default equals method.

[**4 points**] **toString:** Returns a string like "AstroThing(Vega,12.3)" where "Vega" is the name and "12.3" is the distance from earth.

[**2 points**] **main:** A main method that tests each method in this class.

[**4 points**] **Star:** child class of AstroThing

[**3 points**] **constructor:** Has three arguments, a String and a double, for the name and distance from earth, and an int for the first year anything was written about this star. Assigns these values to protected instance variables.

[**3 points**] **getYear:** Returns the year.

[**3 points**] **setYear:** Changes the value of the year.

[**3 points**] **equals:** Returns true if two objects being compared have the same name. Implement this only if the method inherited from AstroThing is not sufficient.

[**3 points**] **toString:** Returns a string like "Star(Vega,25.0, 1840)".

[**2 points**] **main:** A main method that tests each method in this class.

[**4 points**] **Galaxy:** child class of AstroThing

[**3 points**] **constructor:** Has three arguments, a String and a double, for the name and distance from earth, and an int for the diameter of the galaxy in light years. Assigns these values to protected instance variables.

[**3 points**] **getDiameter:** Returns the diameter.

[**3 points**] **setDiameter:** Changes the value of the diameter.

[**3 points**] **equals:** Returns true if two objects being compared have the same name and the same diameter. Implement this only if the method inherited from AstroThing is not sufficient.

[**3 points**] **toString:** Returns a string like "Galaxy(Andromeda,2500000.0, 125000)".

[**2 points**] **main:** A main method that tests each method in this class.

Remember, you are required to implement a `main` method in every class that tests the methods of that class. After your tests in the main methods of each class work, change the main method in your Sky class to perform the following steps.

[**4 points**] Compiles.

1. Add these to your Sky in the order given.
   - The galaxy named Andromeda that is 2,500,000 light years away and has a diameter of 100,000 light years.
   - The star named Vega, that is 26 light years away and first written about in 1215.
   - The star named Sirius that is 8.6 light years away and was first written about in 700 BCE, which you can represent as -700.
   - The star named Altair that is 17 light years away and was written about in 1640.
   - The galaxy named Milky Way that is 0 light years away and has a diameter of 100,000 light years.

[**4 points**] 2. Print the sky.

[**4 points**] 3. Change the value of the distance in the 3rd item in the sky to be twice its current value, by using the get method for ArrayLists get and set methods of AstroThing.

[**1 points**] 4. Print the sky again.

Here are some reminders about how to use ArrayLists.

- Remember to `import java.utils.ArrayList;`
- Constructing an instance: `ArrayList<Type> list = new ArrayList<Type>();`
- Add an object `o` of type Type doing `list.add(o)`
- Get element at index `i` by doing `list.get(i)`

When you are done, combine your files into a **jar file** named `sky.jar`

```
jar cvf sky.jar *.java
```

and check-in your jar file using the Checkin page at CS161 web site.

And you must turn in this exam form with your name on it to your instructor before you leave recitation.