Introduction to Java
Lab Project: 03
Points Possible: 125
Due Date: Oct. 01, 2015 (11:59pm)

Objective: Explore Object Oriented Programming by creating a simple model of our Solar System.

Grade Table:

Jar file is created and is executable	5
Programming Guidelines are followed	5
Report with screenshots is created	25
Planet class is created and meets specs listed below	30
Solar System is created and meets specs listed below	30
Driver class is created and meets specs listed below	30
TOTAL (Points Possible)	125

Instructions: Create a simple command line Java program to model our solar system (use Wikipedia for any data - you do not need to cite, also Pluto's inclusion/exclusion is up to you). To complete this project you will need to create 3 separate java files:

A Planet Class: this file is should be a class that contains 2 private attributes or properties. The first attribute should be for the name and the second a number for the number of moons that the planet has. The class should have a parameterized constructor, "setters and getters", and finally a "toString" method that returns the name and the number of moons.

A Solar System Class: this file should be a class file with a single *private* attribute. The attribute should be an array of planets. The class should have a parameterized constructor that accepts a number for the size of the solar system (the number of planets). Include accessor methods that allow the placing of planets in the array and also a method that allows us to retrieve a planet by its position within the solar system. Additionally include methods that return the number of planets in the solar system and the total number of moons for the solar system. Finally, the Solar System class should have a method that prints out all of the planets it contains and the number of their moons (the toString method for each planet should be called).

A Driver Class: this file should contain a main method that instantiates the solar system and planets and positions the planet objects within the solar system object. Then have the Solar system print itself (that is its planets). Finally print out the total of the moons within the entire solar system.

Note: Only the Driver class should have a main method. Additionally there is no need for user interaction with this program (Scanner or argc). All of the planet data should be hard-coded into the driver class.

Turn In:

1. Create an executable jar file as demonstrated in class (see the class Blackboard site for notes) that contains your source code and class files named "project03LastName.jar"
2. Create a short documentation report (doc or docx) containing screen shots as needed of your program meeting or attempting to meet the specifications in the above grading table. If you are unable to meet various specifications include any error message that are generated when you attempt to compile or run your program. Include a brief description (1 paragraph is fine) documenting your work and describing the functionality of your program. In the report include the version of your Java Compiler (at the command line run "`javac -version`" and any other tools you used.
3. Submit the resulting jar file and documentation report to Blackboard

If you have any questions email me early and often at the below address!

george.patterson@tulsacc.edu