Introduction to Java
Lab Project: 04
Points Possible: 200
Due Date: Oct. 15, 2015 (11:59pm)

Objective: Create a command line flash card program that use Object Oriented Programming principles.

Grade Table:

Jar file is created and is executable!	25
Programming Guidelines are followed	25
Report with screen shots is created	25
FlashCard super class meets specs listed below	25
MathCard sub class meets specs below	25
VocabCard sub class meets specs below	25
CardStack class is created and meets specs below	25
Driver class is created and meets specs listed below	25
TOTAL (Points Possible)	200

Instructions: Create a command line flash card program that utilizes the Object Oriented concepts of *inheritance* and *polymorphism*. To complete this project you will need to create at least 5 separate java files (you may create additional class files as needed):

<u>FlashCard class:</u> This class should be an abstract class and contain at least two abstract methods: getQuestion, and getAnswer. These two methods should return Strings.

<u>MathCard class:</u> This class should inherit from the FlashCard class. Constructors for this class should take three arguments: valueA, mathOperation, and valueB. The values should be integers while the operation should be a String or char. The class should compute the correct answer (valueA mathOperation valueB). This can be done when the object is instantiated or when the getAnswer method is called. Note: your class should deal with *impossible* math (division by zero).

VocabCard class: This class should inherit from the FlashCard class. Constructors for this class should take two String arguments.

CardStack class: This class should have an ArrayList of *Flashcards* and appropriate methods for creating and/or adding new flash cards to the ArrayList. Additionally the class should have a "shuffle" method for randomizing the cards in the ArrayList (this is or should be one line of code – if you are writing a method to randomize your list you are on the wrong track – do some research). Finally, this class needs a nextCard method that will return the next card from the ArrayList. Hint: you may want to keep track of the position of the current card in the CardStack object.

Driver class: This class should instantiate a CardStack object and add 4 VocabCard and 4 MathCards to it. When launched, the Driver should call the CardStacks shuffle method and then display the first card's question. When the "a" key is pressed the card's answer should be shown. When the "n" key is pressed the next card's question should be shown. When the last card's answer has been shown the program should end. Finally, if at any time the "q" key is pressed the program should end. Use a Scanner for input.

Note: Only the Driver class should have a main method. All of the data for the cards should be hard coded in the driver class. Please note that this program is the foundation to future projects that will add both a GUI (Graphic User Interface) and File IO.

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 "project04LastName.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