CSC 120
Programming Project #4

Due Date: ***uploaded to Blackboard by 10:30am*** **Wednesday, March 29, 2017**

Objectives:

- To use the programmer-defined Question class written in Project #3 to write a math game for 1st graders
- To practice writing application code that uses a programmer-defined class
- To add JavaDoc comments to a programmer-defined class
- To add an accessor method to the programmer-defined class Question class

The Problem:

In Project #3, you wrote and tested the **Question** class as the first stage of a project for a school to use to help first grade students practice basic math skills. Recall, the children have learned the following skills:

- addition - two numbers, both in the range 0 - 12
- subtraction - two numbers, the first in the range 6-12 and the second in the range 0 to less than or equal to the first number - for example (12-10, 6-2, 8-0, 7-7, etc.)

In order for this program to be useful, it must be able to generate all possible equations for both operators. Please read the feedback you received on Project #3 and make all necessary modifications to your Question class so that your game can generate all possible equations.

Start Eclipse & Create a New Application Class

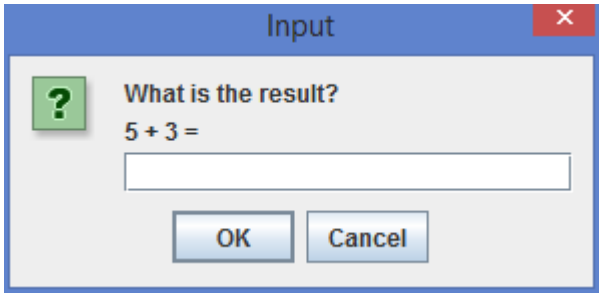
After making any necessary modifications to the ***Question*** class from Project #3, create a new class named ***Project4App*** in the ***proj3*** project that already contains the ***Question*** class. Make sure both the Question class and the Project4App class have a comment at the top with a Title, Description and your name as the author. The Description should explain in a few sentences the purpose of the class as if you were explaining it to someone not familiar with programming. The Question class should contain JavaDoc comments for each constructor and method. Refer to previous labs for the format of JavaDoc comments and add appropriate comments to the Question class.

Writing the application class

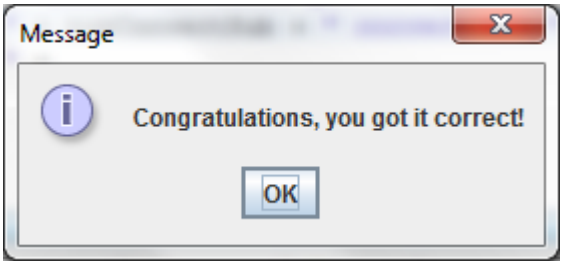
The purpose of the ***Project4App*** class is to create a math game for children to play which will strengthen their addition and subtraction skills. To allow the student to get enough practice, the game will ask the student 10 questions. If the student gets the initial 10 questions correct the game ends. Otherwise, the game continues until one of two things happens: 1) the student's percent of correct answers reaches at least 85% or 2) the student has completed 20 questions (regardless of the percent correct). When the game ends, the student should be told the number of addition questions he/she got correct and incorrect as well as the number of subtraction questions he/she got correct and incorrect. It should also give a score which is the percent of questions correct.

The game should:

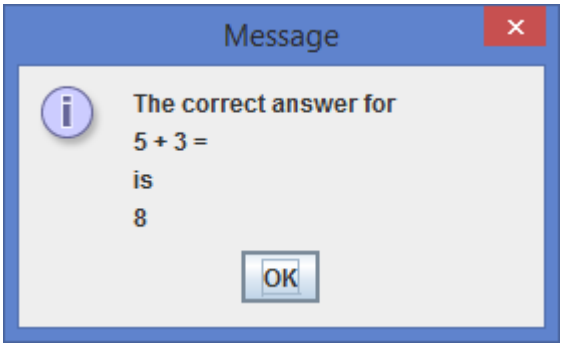
- Display a question in an input dialog box and prompt the user for an answer. For example,



- After the player has typed in an answer and clicked OK, your program should determine if the student has answered the question correctly and provide a message box ***JOptionPane.showMessageDialog(null,"Message to display");*** informing the student if his/her answer is correct. For example,



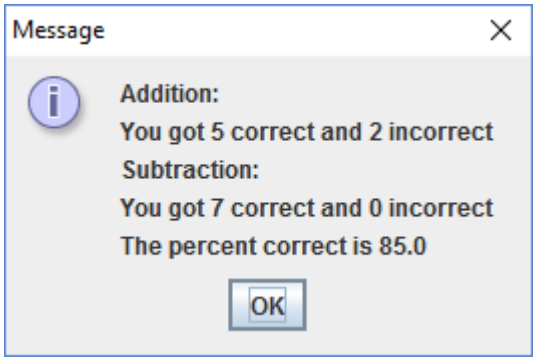
If the answer is incorrect, the message box should show the question and the correct answer. For example,



- The game must keep track of the number of addition questions the student gets correct and incorrect as well as the number of subtraction questions the student answers correctly and incorrectly. Use 4 different counters to keep track of this information. Add an ***accessor*** method to the ***Question*** class that returns the operator so that the application can determine if the current question is addition or subtraction.

- The game should continue until the student has answered the first 10 questions correctly or has reached at least 85% correct or completed 20 questions (see page 1 for a complete description).

- When the game has ended, the student will receive a report of how he/she did. The following MessageDialog is an example of the information that should be provided as feedback - addition: number correct and number incorrect; subtraction: number correct and number incorrect; overall percent correct.



Citation Policy:

This assignment should be worked on individually to help you practice the material covered in class. However, you may work with other students as long as you cite the name(s) of who you worked with and all students understand the assignment submitted. Failure to cite other people is considered plagiarism. You may receive help from your instructor or from the Computer Learning Center in B225.

Grading:

This assignment will be graded in 2 parts, the program will be worth 60 points and the quiz will be worth 40 points.
Part 1: Correctness of both classes (Question and application) of the program submitted. Does the program submitted meet the requirements? Does it display all information? Are the values correct? Have you commented your code?
Part 2: Grade on an in class quiz. On the due date, you will take a 10 minute quiz at the beginning of class based upon the assignment you submit. Bring a printout with line numbers to class with you to use as you take the quiz. The printout must match the program you uploaded to Blackboard and contain your name. Please make sure you are on time for class. Students who are late may not have enough time to complete the quiz or may miss it entirely.

Submission:

Hand in the following in class on March 29th

- The answers to the questions on the sheet attached.
- Printout of the source code for the completed Question.java and Project4.java classes (please include line numbers for the quiz).

Upload the Question.java and Project4.java files for grading by 10:30am on March 29th.

CSC 120
Programming Project #4
Due: Wednesday, March 29th

Name: _____

How many hours did it take you to complete this assignment? (circle one)

- < 1
- 1-3
- 3-5
- 5-7
- > 7

Where did you work on this assignment? (circle all that apply)

- home
- school
- work

How did you go about getting help on this assignment? (circle all that apply)

- talked with an assistant in one of the Learning Centers
- talked with my professor
- talked with another student
- read the textbook
- looked at notes from class
- looked at lab exercises
- other _____ (please specify)

Is there anything that doesn't work as you'd like or expect? If so, what?

What was the most challenging part of writing the code?