CS 1102 - Unit 3 Programming Assignment

In this assignment, you will modify your Quiz class from the previous assignment. You will create a method that asks a quiz question and a method that checks the answer, and you will use those methods to ask multiple questions. Your program will then report how many questions the user got correct. This program will be the basis for future Programming Assignments in this course.

These instructions assume that you are using the Eclipse IDE on your own computer or you may use a different Java environment as long as you can provide Java code and screen shots for your assignment submission.

First open your previous Programming Assignment in Eclipse, with the file "Quiz.java" in the editor pane.

- Add a static method that asks a question until the user provides valid input.

- Call the method "ask". It should take a String parameter, and it should return a String.

  ```
  static String ask(String question) {
  ```

- Remember where method definitions go: inside the class (in this case, "Quiz") but outside all other method definitions (like "main"). It can go before or after the "main" definition.

- Ask the user the question repeatedly until the user provides a valid answer: "A", "B", "C", "D", or "E".

- Ask the question using "JOptionPane.showInputDialog".

- Allow the user to provide a lower-case answer, and convert it to upper case.

- If the user provides an invalid answer, use "JOptionPane.showMessageDialog" to tell them, "Invalid answer. Please enter A, B, C, D, or E." Then ask the question again. Repeat until the answer is valid.

- Once the user provides a valid answer, *return* that answer (converted to upper case).

- Notice that the method does not check whether the answer is correct. It just asks the same question until the answer is *valid*.

Try out your new method.

- Delete the existing loop from your main method, but leave the initialization of the question String.

- Call the "ask" method with your question String as the argument (actual parameter).

- Note that you do not have to use the String returned by "ask".

- Run your program. It should keep asking the question until you provide a valid answer. It does not yet check the answer.

- Try invalid and valid answers.

Now add a method that asks questions using "ask" and checks answers.

- Call the method "check". It should take two String parameters. `static void check(String question, String correctAnswer) {`

- The method definition can go before, after, or between the definitions for "ask" and "main", but it must be inside the "Quiz" class and outside the other method definitions.

- In the "check" method, call "ask" to get a valid answer. `String answer = ask(question);`

- If the answer is correct, use "JOptionPane.showMessageDialog" to report, "Correct!"

- If the answer is incorrect, use "JOptionPane.showMessageDialog" to report, "Incorrect. The correct answer is ." This output should include the actual correct answer.

Test the "check" method.

- Replace the "ask" call in the main method with a "check" call. You will need to provide both the question String and the answer String as arguments.

- Run your program. It should keep asking the question until you provide a valid answer. Then it should tell you if your answer is correct.

- Try invalid, incorrect, and correct answers.

Add more questions.

- In the main method, call "check" with at least two more unique quiz questions, each with a different correct answer, for a total of at least three questions.

- You may either declare and initialize new local variables or reassign the existing question variable after each "check" call.

- Run your program. Make sure the responses work as expected for all the questions.

Finally, add a score for the quiz.

- Add two static member variables, one for the number of questions and one for the number of correct answers. Initialize them to zero for good documentation.
  ```
  static int nQuestions = 0;
  static int nCorrect = 0;
  ```

- Remember where member variables go: inside a class definition but outside all method definitions. You may declare the variables before, after, or between the existing methods.

- In the "check" method, increment "nQuestions" each time it is called.

- Also in the "check" method, increment "nCorrect" for each correct answer.

- Display the score at the end of the main method using "JOptionPane.showMessageDialog". Use the text, " correct out of questions", with the appropriate numbers.

- Run your program. Make sure it displays the right numbers of questions and correct answers.