Unit 4 Programming Assignment

In this assignment, you will again modify your Quiz program from the previous assignment. You will create a separate class for quiz questions, and you will create objects of that class to ask questions and check answers.

This assignment will include multiple cut-and-paste operations from the existing "Quiz" class into the new "MultipleChoiceQuestion" class. Object-oriented programming is designed to avoid cut and paste, and you will see some of the techniques for re-using existing code in the next assignment. In this assignment, however, you will be converting from procedural programming to object-oriented programming, and cut-and-paste is a simple strategy for this conversion.

First create the new "MultipleChoiceQuestion" class.

- Open your "CS1102" project in Eclipse.
- Select "New" -> "Class" from the File menu.
- Enter the Name "MultipleChoiceQuestion" and click "Finish".

The new file "MultipleChoiceQuestion.java" should appear in the editor pane. Make sure it is also listed under "(default package)" in the "Package Explorer" pane, along with "Quiz.java". Having both files in the same package eliminates complications with package imports and access specifiers when the "Quiz" class tries to use the "MultipleChoiceQuestion" class.

Add <u>class variables</u> and <u>instance variables</u> to the "MultipleChoiceQuestion" definition.

- Copy the class variables "nQuestions" and "nCorrect" from the "Quiz" class and paste them into the "MultipleChoiceQuestion" class.
  ```
  static int nQuestions = 0;
  static int nCorrect = 0;
  ```
- Add instance variables for the question and correct answer. Like class variables, these go inside the class definition for "MultipleChoiceQuestion".
  ```
  String question;
  String correctAnswer;
  ```

Now add a <u>constructor</u> for "MultipleChoiceQuestion" objects.

- Name the constructor "MultipleChoiceQuestion". Remember that constructors have no return type. Their name is the return type!
- Give the constructor 7 parameters, all of type String, named the following
  - query
  - a, b, c, d, and e
  - answer
- Thus, the constructor should start as follows.
  ```
  MultipleChoiceQuestion(String query, String a, String b, String
  c, String d, String e, String answer) {
  ```

(Notice that the parameter names for the constructor are all different from the instance variable names so that they can be distinguished. In the next unit, you will learn how to distinguish between parameters and instance variables with the same name.)

Have the constructor initialize the instance variables using its parameters.

- Initialize "question" using the "query" parameter followed by each choice parameter, "a"-"e".
- Remember to add the letter and a newline for each choice. For example:
  ```
  question = query+"\n";
  question += "A. "+a+"\n";
  ...
  ```
- Initialize "correctAnswer" to the parameter "answer". Convert it to upper case, so that the String "a" will work as an answer, in addition to "A".

Convert "ask" from a class method of "Quiz" to an instance method of "MultipleChoiceQuestion".

- Add the import statement for "JOptionPane" to "MultipleChoiceQuestion.java" since the copied methods call "JOptionPane" methods.
- Copy the "ask" method from the "Quiz" class and paste it into the "MultipleChoiceQuestion" class.
- Remove the "static" modifier and the "question" parameter, leaving the following instance method with no parameters. It does not need parameters because it will use the instance variables.
  ```
  String ask() {
  ```
- If you named your variables as instructed in the last assignment, the "ask" method should now use the instance variable "question" without further modification. Otherwise, edit the method to use "question".

Now convert "check" from a class method of "Quiz" to an instance method of "MultipleChoiceQuestion".

- Copy the "check" method from the "Quiz" class and paste it into the "MultipleChoiceQuestion" class.
- Remove the "static" modifier and the two parameters from "check", leaving the following instance method with no parameters. It does not need parameters because it will use the instance variables.
  ```
  void check() {
  ```
- Remove the "question" parameter from the "ask" call in "check". Both methods have direct access to the instance variable.
- If you named your variables as instructed in the last assignment, the "check" method should not require further modification. Otherwise, edit the method to use the correct names for instance variables and class variables.

Add a new class method called "showResults" to display the number of questions and correct answers.

- Define the class method "showResults". It needs no parameters and returns no value.
  ```
  static void showResults() {
  ```
- Copy the "JOptionPane" call that displays the results from the main method of "Quiz". Paste it into the "showResults" method.
- Again, if you named your variables as instructed, this method should not require further modification. Otherwise, edit the method to use the correct names for the class variables.

Your "MultipleChoiceQuestion" class should now be ready to use. Make sure that no error icons appear on the left side of the editor pane for "MultipleChoiceQuestion.java". If they do appear, mouse over them to see what needs to be fixed.

Next convert the "Quiz" class to use the "MultipleChoiceQuestion" class.

First delete the following from "Quiz.java".

- The "import" statement.
- The class variables.
- The class methods "ask" and "check".

Convert the main method to construct and use "MultipleChoiceQuestion" objects.

- Instead of using a String for "question", create "MultipleChoiceQuestion" objects. Here is an example.
  ```
  MultipleChoiceQuestion question = new MultipleChoiceQuestion(
      "What is a quiz?",
      "a test of knowledge, especially a brief informal test given
  to students",
      "42",
      "a duck",
      "to get to the other side",
      "To be or not to be, that is the question.",
      "a");
  ```
  Notice that the constructor call does not need the choice letters ("A"-"E") or newlines because the constructor adds those automatically. Also notice that the correct answer can be upper or lower case because the constructor automatically converts it to upper case.
- Replace calls to "check(question)", where "question" is a String, with calls to "question.check()", where "question" is a MultipleChoiceQuestion.
- Replace the message dialog that shows the results with a call to the class method "showResults".

Note that you can either use a new MultipleChoiceQuestion variable for each question or you can use the same one and just reset its value to reference another object created with "new". Java garbage collection will clean up the finished objects.

Test your program with the same questions from the previous assignment.

Finally, add at least two more questions, for a total of at least five. Test your program to make sure it asks the questions and responds appropriately to correct, incorrect, and invalid answers.

Your assignment will be graded by your peers using the following criteria. Each item is worth 10 points, for a total of 90 points.

- Does the submission include a file that defines the class "Quiz"?
- Does the submission include a file that defines the class "MultipleChoiceQuestion"?
- Does the class "MultipleChoiceQuestion" have class variables for the number of questions and the number of correct answers and instance variables for the question String and correct-answer String?
- Does the class "MultipleChoiceQuestion" have a constructor with 7 String parameters that it uses to initialize the instance variables?
- Does the class "MultipleChoiceQuestion" have a method "ask" that asks the question and returns a valid answer in upper case?
- Does the class "MultipleChoiceQuestion" have a method "check" that calls "ask" and displays one message for correct answers and a different message for incorrect answers?
- Does the class "MultipleChoiceQuestion" have a class method "showResults" that displays the number of questions and the number of correct answers using the class member variables?
- Does the main method in class "Quiz" use the "MultipleChoiceQuestion" constructor to create at least 5 quiz questions?
- Does the main method in class "Quiz" use a "MultipleChoiceQuestion" object to call the "check" method for each quiz question, and an object or the "MultipleChoiceQuestion" class to call the "showResults" class method?

Upload the following for your Programming Assignment submission.

- Quiz.java
- MultipleChoiceQuestion.java
- Screen shot showing the input dialog with a new quiz question, not one from the previous assignment
- Screen shot showing the message dialog for the number of questions and correct answers