

CPSC 1181 - Lab 3 [50 marks]

Objectives:

- Create an ArrayList of objects
- Explain how programming instructions are executed when several classes are involved
- Describe the difference between a tester class/driver program/test case
- Implement static methods and variables
- Implement properly overloaded constructors

Instructions:

- Complete all of the exercises and submit your zip file prior to the due date.

Submission:

- Zip up your Question.java, Quiz.java, TestQuestion.java, and QuizTime.java and submit them to D2L.
- Unzipped submissions or submissions containing .class files will be automatically given ZERO

Exercise 1

Create a class called Question that simulates short-answer style questions to be placed on a quiz.

Each Question object should have a question, an answer, and a difficulty level. The difficulty level should be an integer between 1 and 10 that represents the difficulty level of the question (1 being easy, 10 being hard).

Overload the constructor so that it only accepts 2 parameters, the question and answer, and sets the difficulty to a default value of 1. Make sure to use the `this` keyword.

Create the getters and setters for the instance data of this class. Create a `toString` method that returns a string summary of the Question object.

Create a method `boolean isCorrect(String attemptedAnswer)` that returns true if the attempted answer is the correct answer for the current question.

Exercise 2

Download and run the TestQuestion and ensure that your Question class passes all of the JUnit tests.

Exercise 3

Create a class called Quiz that uses your Question class to create Quiz objects.

A Quiz can be composed of *up to* 25 questions and a score. The Quiz constructor should accept the number of questions on the quiz as a parameter and initialize an ArrayList accordingly. Create a way to keep track of the total number of quizzes ever created.

Create a method called `void add(Question q)` that adds a new question to the quiz. What should happen if we try to add a 26th question to a quiz?

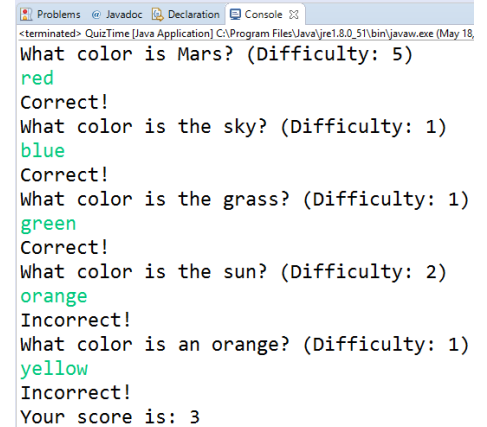
Create **only the necessary** getters and setters.

Create a method called `void giveQuiz()` that presents each question, and its difficulty, in turn to the user, accepts an answer for each one, and keeps track of the results.

CPSC 1181 - Lab 3 [50 marks]

Exercise 4

Create a tester class (**read: not a JUnit test class**) called QuizTime that creates and populates a Quiz, presents it, and prints the final results. Pick an interesting topic to make your quiz about.
To the right is a Sample Output for one execution of QuizTime



```
<terminated> QuizTime [Java Application] C:\Program Files\Java\jre1.8.0_51\bin\javaw.exe (May 18, 2016)
What color is Mars? (Difficulty: 5)
red
Correct!
What color is the sky? (Difficulty: 1)
blue
Correct!
What color is the grass? (Difficulty: 1)
green
Correct!
What color is the sun? (Difficulty: 2)
orange
Incorrect!
What color is an orange? (Difficulty: 1)
yellow
Incorrect!
Your score is: 3
```

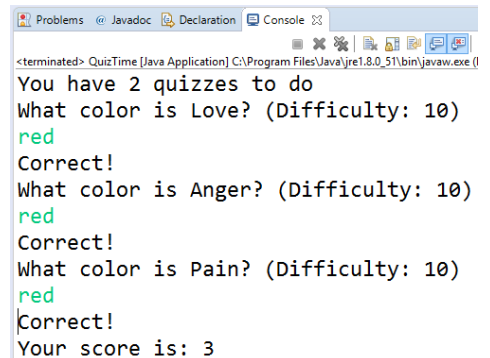
Exercise 5

Currently the difficulty of the questions is not used in creating the quizzes. Overload the giveQuiz method in the Quiz class so that it accepts two integer parameters that specify the minimum and maximum difficulty levels for the quiz questions and presents only questions in that difficulty range.

Hint: work smarter not harder, don't duplicate code. If your overridden method contains almost all of the same lines of code as the original giveQuiz method, you're doing it wrong.

Create another Quiz in your tester class that demonstrates this new feature. Also demonstrate that you successfully track the total number of quizzes created.

To the right is a Sample Output for one execution of QuizTime



```
<terminated> QuizTime [Java Application] C:\Program Files\Java\jre1.8.0_51\bin\javaw.exe (May 18, 2016)
You have 2 quizzes to do
What color is Love? (Difficulty: 10)
red
Correct!
What color is Anger? (Difficulty: 10)
red
Correct!
What color is Pain? (Difficulty: 10)
red
Correct!
Your score is: 3
```

Marking Rubric:

Style, Convention, Documentation [5 marks]

Question.java [9 marks]

- +1.5 instance data
- +2 constructor
- +2 overload constructor correctly
- +2.5 get/set question/answer/difficulty
- +1 toString

TestQuestion.java [7 marks]

- +3 for passing getters and setters tests
- +1 for passing each other method test

Quiz.java [22 marks]

- +2 instance data
- +1 tracking total quizzes
- +5 constructor
- +2 for getting total quizzes correctly
- +2 addQuestion
- +1 for getting score correctly
- +7 for give quiz
- +2 for correctly overloading give quiz

QuizTime.java [7 marks]

- +3 for creating a quiz, filling it and running it
- +3 for creating a second quiz, filling and running it
- +1 for showing total number of quizzes