**Langara College**

**Department of Computing Science & Information Systems**

**CPSC1181 – Object-oriented Computing**

###### **Lab4: More Classes and ArrayLists**

**Objectives:**

* Create an ArrayList of objects
* Explain how programming instructions are executed when several classes are involved
* Implement overloaded methods
* Implement and use an enumerated type

**Instructions:**

1. Open Eclipse and make **LabProjects** as the workspace.
2. Inside Eclipse, create a project named **Lab4** to store all the files for this lab. Make sure uncheck **Create module-info.java file**.
3. Create a **src** folder and then a package named **lab4** inside the project **Lab4.** Store all your source files inside the package **lab4**
4. Add Javadoc comments and regular comments to all your code

**Problem [55 marks]**

**Part I: [15 Marks] The Question class**

1. Create a class called Question that simulates short-answer style questions to be placed on a quiz.
2. Each Question object should have a question, an answer, and a difficulty level. The difficulty level should be set using an enum defined inside the Question class. The enum should have at least three named values ranging from easiest to hardest.
3. Create a constructor that accepts the question, the answer, and the difficulty.
4. Create a constructor that accepts only the question and answer, and sets the difficulty to a default value of your choosing. Be sure to use the this keyword to use the first constructor.
5. 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.
6. Create a copy method, Question copy() that makes a copy of the current question. For example, q1.copy() returns a Question with the same question, answer and difficulty as q1.

**Part II: [10 marks] Test the Question class**

Create a JUnit test class to ensure that all of the methods (except toString) of your Question class work correctly.

**Part III: [15 marks] The QuizMaker class**

* 1. Create a class called QuizMaker that uses your Question class to create quizzes. A quiz will just be an ArrrayList of questions. A QuizMaker should have a name and a pool of questions (ArrayList) that is initially empty.
  2. Create a constructor that sets the name and creates a new empty ArrayList of questions.
  3. Create a method, void add(Question q) that adds a new question to the pool. Make a copy of the parameter to copy into the pool Arraylist.
  4. Create a getter and setter for the name only.
  5. Create two versions of a method, ArrayList<Question> createQuiz that creates a new ArrayList of questions that will be given as a quiz.
* One which takes no parameters and returns a copy of all questions in the pool.
* One which takes a difficulty value and returns a copy of all questions with that specific difficulty level.

When returning the ArrayList of questions, **you must copy the questions** from the pool using the copy method and not just add them directly. Otherwise, we’re giving away references to the questions in the pool.

It is ok if the method returns an empty ArrayList when there are no questions or no questions at the appropriate difficulty level.

**Part IV: [15 marks] The QuizManager class**

Create a QuizManager class (a manual test class) that includes…

* + 1. A method static void giveQuiz(ArrayList<Question> questions) that presents each question in the parameter ArrayList in turn to the user, accepts an answer for each one, and displays the score at the end. Use a Scanner to read user input.

Note: If the parameter ArrayList is empty it should throw an exception

* + 1. A main method that creates a QuizMaker object and adds at least 10 questions including questions of all of the possible difficulty levels. Your questions can be simple, but have them make sense. **Up to 2 bonus marks** for creating an interesting set of questions. It then should use QuizMaker’s createQuiz methods to create two quizzes. One that contains all questions and the other with only the questions at a particular difficulty level. Then run those quizzes using the giveQuiz method.
    2. This class doesn’t need a constructor, toString, or anything like that.

**What to hand in**

1. Click Project->Generate Javadoc… to geneate the Javadoc documents for all the java files inside Lab4->src->lab4. Leave all the other selections unchanged.
2. Zip the folder **Lab4** and upload it to D2L.

**When to hand in**

By 11:59pm, Wednesday, February 1, 2023.