

**Due** Oct 26 by 11:59pm      **Points** 100      **Available** after Oct 16 at 8am

## Homework 5

 Home

## Modules

[illegible]

## Homework 5: Questions in a questionnaire

## Questions in a questionnaire

---

Many online questionnaire tools like SurveyMonkey, Doodle Poll, etc. (even Blackboard) allow creating a questionnaire made of several types of questions: Yes/No, Short-answer, Likert scale, etc. In this assignment, you will write classes that represent different types of questions, and a class to represent a questionnaire.

Each question, irrespective of type, has the following common aspects:

- It has the text of the question itself, and a method `getPrompt` that returns it.
- It is either required or optional, and has a method `isRequired` that returns its status.
- It has a method `answer (String)` that allows one to enter an answer as a `String`. What the string may contain depends on the type of question.
- It has a method `getAnswer` that returns the answer to the question, or empty string if there is no answer.
- It has a method `copy` that returns a copy of the question including all its data.

The types of questions are:

1. **YesNo**: this can be answered in one of two ways: yes or no. `answer (String)` would accept "Yes" or "No" as valid answers, but case-insensitive, so "yes" or "NO" would also be valid.
2. **ShortAnswer**: this can be answered in at most 280 characters, including spaces.
3. **Likert**: this can be answered on a fixed, 5-point Likert scale (Strongly Agree, Agree, Neither Agree nor Disagree, Disagree, Strongly Disagree). `answer (String)` would accept only these precise words as valid answers, but again case-insensitive.

All code for this assignment should be in the `questionnaire` package.

Design an interface `Question` to represent a question, with the methods listed. Then design the implementing classes `YesNo`, `ShortAnswer`, and `Likert`. Each of those classes should have a constructor that takes in the question prompt as a `String` and a `boolean` where `true` means the question is required, and `false` means optional. The `answer` method in each of these classes will enforce the answer requirements for that question type. Consider using the `equalsIgnoreCase ()` method on the `String` class to check for case-insensitive `String` equality.

We have supplied you with an `enum` definition representing the Likert response options. (Not all aspects of this enum are relevant for this assignment.) Note that you can get the values of an enum as an array using the static `values()` method on the class, such as `LikertResponseOption.values()`.

We have supplied you with an interface `Questionnaire`, representing a collection of questions. Implement this interface in a class called `QuestionnaireImpl`, that has only a no-argument constructor. Within this `QuestionnaireImpl` class, you must make use of Java's built-in `List<T>` interface and a built-in implementation of `List<T>`.

Write tests for both the `Question` and the `Questionnaire` interface, to ensure that they work correctly in a range of situations. Do not modify any of the given code (except for style fixes should they be necessary).

You are required to submit a UML class diagram for this assignment - remember to include it with your submission.

**Starter code:** [LikertResponseOption.java](#) ↓

[https://northeastern.instructure.com/courses/123246/files/16688156/download?download\\_frd=1](https://northeastern.instructure.com/courses/123246/files/16688156/download?download_frd=1)

[Questionnaire.java](#) ↓ [https://northeastern.instructure.com/courses/123246/files/16688161/download?download\\_frd=1](https://northeastern.instructure.com/courses/123246/files/16688161/download?download_frd=1)

## How to submit

---

1. You can upload your homework to Gradescope by create a zip file that contains the src and test directories (with the appropriate package directories as specified in this assignment)

OR

2. You can submit to Gradescope via your GitHub account if you have one. As part of your academic honesty agreement, ensure your GitHub repo is **private**, not public!