CS 5010 Fall 2021 Section 8 (Vancouver) (/courses/152)

Fall 2021 Section 19263 : Coria Mendoza at M 6:30 PM - 8:30 PM / W 4:00 PM - 6:00 PM (PT) (Lecture)

You are a **student** of this course.

Teams (/courses/152/teamsets) | Assignments (/courses/152/assignments)

This will be an individual response.

Question.java

```
1  package questions;
2
3  /**
4   * This interface represents a general question on a computer-based test.
5   */
6  public interface Question extends Comparable<Question> {
7
8    final String CORRECT = "Correct";
9    final String INCORRECT = "Incorrect";
10
11   /**
12    * Determines if the answer is correct for a given question. If the answer
13    * correct, this method returns "Correct"; and "Incorrect" otherwise.
14    *
15    * @param answer the answer given
16    * @return "Correct" or "Incorrect"
17    */
18   String answer(String answer);
19
20   /**
21    * Returns the text of the question.
22    *
23    * @return the text
24    */
25   String getText();
26 }
27
```

Implementation

1. Did you document your classes with a comment that describes what the class represents (not how the class is implemented) using correct Javadoc formatting? (1 point)

   Yes | No

   Tag one class Javadoc block where you did this, if it exists

   File: -- select a file -- Line number:

   Optional explanation

2. Does your implementation add any methods to the provided `Question` interface? (1 point)

   Yes | No

   Tag one of the extra methods

   File: -- select a file -- Line number:

   Tag another of the extra methods if it exists

   File: -- select a file -- Line number:

   Optional explanation

3. Does your implementation have public methods that are not part of the interface? (1 point)

Yes No

Tag one of the extra methods

File: -- select a file -- Line number:

Tag another of the extra methods

File: -- select a file -- Line number:

Tag another of the extra methods

File: -- select a file -- Line number:

Optional explanation

4. How does your implementation capture the similarities between questions to avoid code duplication? (3 points)

Tag a related line that supports your response

File: -- select a file -- Line number:

Tag a related line that supports your response

File: -- select a file -- Line number:

Optional explanation

5. Does your implementation use double dispatch to implement `compareTo` ? (3 points)

Yes | No

Tag the first line of the `compareTo` method

File: -- select a file -- Line number:

Tag the first line of the method for comparing True/False

File: -- select a file -- Line number:

Tag the first line of the method for comparing multiple choice

File: -- select a file -- Line number:

Tag the first line of the method for comparing multiple select

File: -- select a file -- Line number:

Tag the first line of the method for comparing Likert

File: -- select a file -- Line number:

Optional explanation

6. In a point list, describe the changes that would need to be made to your implementation if the order between questions were changed (but no new types were added). Refer to any tagged lines that help us understand your explanation. (2 points)

Tag 1

File: -- select a file -- Line number:

Tag 2

File: -- select a file -- Line number:

Tag 3

File: -- select a file -- Line number:

Tag 4

File: -- select a file -- Line number:

7. Does your implementation override the `equals` and `hashcode` methods? (1 point)

   - ◯ No, my implementation does not override either method
   - ◯ My implementation overrrides the `equals` method but not `hashcode`
   - ◯ Yes, my implementation overrides both of these methods

   Tag one of the `equals` methods

   File: -- select a file -- Line number:

   Tag one of the `hashcode` methods

   File: -- select a file -- Line number:

   Optional explanation

Testing

8. Do you have at least one test that verifies a True/False question behaves correctly when given the correct answer? (1 point)

   Yes No

   Tag the first relevant JUnit assert or the expected exception

   File: -- select a file -- Line number:

9. Do you have at least one test that verifies a True/False question behaves correctly when given the incorrect answer? (1 point)

   Yes No

Tag the first relevant JUnit assert or the expected exception

File: -- select a file -- Line number:

10. Do you have at least one test that verifies a multiple choice question behaves correctly when given the correct answer? (1 point)

Yes No

Tag the first relevant JUnit assert or the expected exception

File: -- select a file -- Line number:

11. Do you have at least one test that verifies a multple choice question behaves correctly when given the incorrect answer? (1 point)

Yes No

Tag the first relevant JUnit assert or the expected exception

File: -- select a file -- Line number:

12. Do you have at least one test that verifies a multiple select question behaves correctly when given the correct answer? (1 point)

Yes No

Tag the first relevant JUnit assert or the expected exception

File: -- select a file -- Line number:

13. Do you have at least one test that verifies a multiple select question behaves correctly when given the incorrect answer? (1 point)

Yes No

Tag the first relevant JUnit assert or the expected exception

File: -- select a file -- Line number:

14. Do you have at least one test that verifies a Likert question behaves correctly when given the correct answer? (1 point)

Yes No

Tag the first relevant JUnit assert or the expected exception

File: -- select a file -- Line number:

15. Do you have at least one test that verifies a Likert question behaves correctly when given the incorrect answer? (1 point)

Yes No

Tag the first relevant JUnit assert or the expected exception

File: -- select a file -- Line number:

16. Do you have at least one test that verifies that True/False questions are placed before all other known types of questions in sorted order? (1 point)

Yes No

Tag the first relevant JUnit assert or the expected exception

File: -- select a file -- Line number:

17. Do you have at least one test that verifies that multiple choice questions are placed after the True/False questions but before the multiple select questions in sorted order? (1 point)

Yes No

Tag the first relevant JUnit assert or the expected exception

File: -- select a file -- Line number:

18. Do you have at least one test verifies that multiple select questions are placed after all the multiple choice questions but before the Likert questions in sorted order? (1 point)

Yes No

Tag the first relevant JUnit assert or the expected exception

File: -- select a file -- Line number:

19. Do they have a test that verifies that an array or collection that contains your question objects is sorted correctly using `Arrays.sort` or `Collection.sort` ? (1 point)

Yes No

Tag the first relevant JUnit assert or the expected exception

File: -- select a file -- Line number:

Student notes

Submit response