```java
package il.ac.hit.quizzy;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import java.util.List;
import javax.swing.*;

public class Quiz implements IQuiz {
    private String name;
    private List<IQuizQuestion> questions = new ArrayList<>();

    @Override
    public void start() {
        // Create and show the quiz selection screen using SwingUtilities.invokeLater()
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                showPlatformSelection();
            }
        });
    }

    @Override
    public void setName(String text) {
        this.name = text;
    }

    @Override
    public String getName() {
        return name;
    }

    @Override
    public void addQuestion(IQuizQuestion question) {
        questions.add(question);
    }

```

```
39      @Override
40      public List<IQuizQuestion> getQuestions() {
41          return questions;
42      }
43
44      // Method to show the platform selection screen
45      private void showPlatformSelection() {
46          JFrame frame = new JFrame("Quiz Platform
    Selection");
47          frame.setDefaultCloseOperation(JFrame.
    EXIT_ON_CLOSE);
48          frame.setSize(300, 150);
49          frame.setLayout(new FlowLayout());
50
51          JButton terminalButton = new JButton("
    Terminal Quiz");
52          JButton guiButton = new JButton("GUI Quiz"
    );
53
54          frame.add(terminalButton);
55          frame.add(guiButton);
56
57          terminalButton.addActionListener(new
    ActionListener() {
58              public void actionPerformed(ActionEvent
     e) {
59                  // Start the terminal-based quiz
60                  IQuiz terminalQuiz =
    createTerminalQuiz();
61                  terminalQuiz.start();
62                  frame.dispose();
63              }
64          });
65
66          guiButton.addActionListener(new
    ActionListener() {
67              public void actionPerformed(ActionEvent
     e) {
68                  // Start the GUI-based quiz
69                  IQuiz guiQuiz = createGUIQuiz();
70                  guiQuiz.start();
```

```
 71                  frame.dispose();
 72              }
 73          });
 74
 75          // Center the JFrame on the screen
 76          frame.setLocationRelativeTo(null);
 77
 78          frame.setVisible(true);
 79      }
 80
 81      // Method to create a terminal-based quiz
 82      private IQuiz createTerminalQuiz() {
 83          // Implement logic to create a terminal-
    based quiz with questions loaded from a file
 84          IQuizFilesDAO dao = SimpleCSVQuizFilesDAO.
    getInstance();
 85          try {
 86              IQuiz loadedQuiz = dao.
    loadQuizFromFile("quiz1.data");
 87
 88              // Create a TerminalQuiz instance and
    set the questions
 89              return new TerminalQuiz("Quiz Demo",
    loadedQuiz.getQuestions());
 90          } catch (QuizException e) {
 91              e.printStackTrace();
 92              // Handle the exception as needed
 93          }
 94          return null;
 95      }
 96
 97      // Method to create a GUI-based quiz
 98      private IQuiz createGUIQuiz() {
 99          // Implement logic to create a GUI-based
    quiz with questions loaded from a file
100          IQuizFilesDAO dao = SimpleCSVQuizFilesDAO.
    getInstance();
101          try {
102              IQuiz loadedQuiz = dao.
    loadQuizFromFile("quiz1.data");
103
```

```
104             // Create a GUIQuiz instance and set
    the questions
105             return new GUIQuiz("Quiz Demo",
    loadedQuiz.getQuestions());
106         } catch (QuizException e) {
107             e.printStackTrace();
108             // Handle the exception as needed
109         }
110         return null;
111     }
112 }
113
```

```java
package il.ac.hit.quizzy;

import java.util.List;

public interface IQuiz {
    void start();
    void setName(String text);
    String getName();
    void addQuestion(IQuizQuestion question);
    List<IQuizQuestion> getQuestions();
}

```

```java
package il.ac.hit.quizzy;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.List;

public class GUIQuiz implements IQuiz {
    private String name;
    private List<IQuizQuestion> questions;
    private JFrame frame;

    public GUIQuiz(String name, List<IQuizQuestion> questions) {
        this.name = name;
        this.questions = questions;
    }

    @Override
    public void start() {
        createAndShowGUI();
    }

    private void createAndShowGUI() {
        frame = new JFrame("Quiz GUI");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 400);

        // Center the JFrame on the screen
        frame.setLocationRelativeTo(null);

        JPanel quizPanel = new JPanel();
        quizPanel.setLayout(new BoxLayout(quizPanel, BoxLayout.Y_AXIS));

        for (IQuizQuestion question : questions) {
            JLabel questionLabel = new JLabel(question.getQuestion());
            quizPanel.add(questionLabel);
```

```
38
39              List<IQuizAnswer> answers = question.
   getAnswers();
40              ButtonGroup buttonGroup = new
   ButtonGroup();
41              for (IQuizAnswer answer : answers) {
42                  JRadioButton radioButton = new
   JRadioButton(answer.getText());
43                  buttonGroup.add(radioButton);
44                  quizPanel.add(radioButton);
45
46                  // Add action listener to each
   radio button
47                  radioButton.addActionListener(new
   ActionListener() {
48                      @Override
49                      public void actionPerformed(
   ActionEvent e) {
50                          answer.setSelected(
   radioButton.isSelected());
51                      }
52                  });
53              }
54          }
55
56          JButton submitButton = new JButton("Submit"
   );
57          submitButton.addActionListener(new
   ActionListener() {
58              @Override
59              public void actionPerformed(ActionEvent
    e) {
60                  calculateAndDisplayGrade();
61              }
62          });
63
64          quizPanel.add(submitButton);
65
66          frame.add(new JScrollPane(quizPanel));
67
68          // Center the JFrame on the screen after
```

```
 68 adding components
 69         frame.setLocationRelativeTo(null);
 70 
 71         frame.setVisible(true);
 72     }
 73 
 74     private void calculateAndDisplayGrade() {
 75         int totalQuestions = questions.size();
 76         int correctAnswers = 0;
 77 
 78         for (IQuizQuestion question : questions) {
 79             List<IQuizAnswer> answers = question.
    getAnswers();
 80             for (IQuizAnswer answer : answers) {
 81                 if (answer.isSelected() && answer.
    isCorrect()) {
 82                     correctAnswers++;
 83                 }
 84             }
 85         }
 86 
 87         double percentage = (double)
    correctAnswers / totalQuestions * 100;
 88         String message = "You got " +
    correctAnswers + " out of " + totalQuestions + "
    correct.\n";
 89         message += "Your score: " + percentage +
    "%";
 90         JOptionPane.showMessageDialog(frame,
    message, "Quiz Results", JOptionPane.
    INFORMATION_MESSAGE);
 91         frame.dispose();
 92     }
 93 
 94     @Override
 95     public void setName(String text) {
 96         this.name = text;
 97     }
 98 
 99     @Override
100     public String getName() {
```

```
101             return name;
102         }
103
104         @Override
105         public void addQuestion(IQuizQuestion question
    ) {
106             questions.add(question);
107         }
108
109         @Override
110         public List<IQuizQuestion> getQuestions() {
111             return questions;
112         }
113 }
114
```

```java
package il.ac.hit.quizzy;

public class Program {

    public static void main(String[] args) throws QuizException {
        // Creating a quiz
        QuizFactory factory = new QuizFactory();
        IQuiz quiz = factory.createQuiz(QuizType.GUI);
        quiz.setName("Quiz Demo");

        // Creating the 1st question
        IQuizQuestionBuilder builder1 = new QuizQuestion.Builder();
        builder1.setTitle("We Love Canada");
        builder1.setQuestion("Canada starts with…?");
        builder1.addAnswer("Canada starts with the letter 'A'.", false);
        builder1.addAnswer("Canada starts with the letter 'B'.", false);
        builder1.addAnswer("Canada starts with the letter 'C'.", true);
        builder1.addAnswer("Canada starts with the letter 'D'.", false);
        builder1.addAnswer("Canada starts with the letter 'E'.", false);
        IQuizQuestion question1 = builder1.create();

        // Creating the 2nd question
        IQuizQuestionBuilder builder2 = new QuizQuestion.Builder();
        builder2.setTitle("We Love Australia");
        builder2.setQuestion("Australia starts with …?");
        builder2.addAnswer("Australia starts with the letter 'A'.", true);
        builder2.addAnswer("Australia starts with the letter 'B'.", false);
```

```
28          builder2.addAnswer("Australia starts with
   the letter 'C'.", false);
29          builder2.addAnswer("Australia starts with
   the letter 'D'.", false);
30          builder2.addAnswer("Australia starts with
   the letter 'E'.", false);
31          IQuizQuestion question2 = builder2.create
   ();
32
33          // Adding questions to the quiz
34          quiz.addQuestion(question1);
35          quiz.addQuestion(question2);
36
37          // Saving the quiz to a file and reading it
    back
38          IQuizFilesDAO dao = SimpleCSVQuizFilesDAO.
   getInstance();
39          dao.saveQuizToFile(quiz, "quiz1.data");
40          IQuiz loadedQuiz = dao.loadQuizFromFile("
   quiz1.data");
41          loadedQuiz.start();
42      }
43  }
44
```

```java
package il.ac.hit.quizzy;

public enum QuizType {
    TERMINAL, GUI
}

```

```java
package il.ac.hit.quizzy;

public class QuizAnswer implements IQuizAnswer {
    private String text;
    private boolean correct;
    private boolean selected; // Add selected property

    public QuizAnswer(String text, boolean correct) {
        this.text = text;
        this.correct = correct;
    }

    @Override
    public String getText() {
        return text;
    }

    @Override
    public boolean isCorrect() {
        return correct;
    }

    @Override
    public boolean isSelected() {
        return selected;
    }

    @Override
    public void setSelected(boolean selected) {
        this.selected = selected;
    }
}

```

```java
package il.ac.hit.quizzy;

public interface IQuizAnswer {
    String getText();
    boolean isCorrect();
    boolean isSelected();
    void setSelected(boolean selected);
}

```

```java
package il.ac.hit.quizzy;

import java.util.ArrayList;
import java.util.List;

public class QuizFactory {
    public IQuiz createQuiz(QuizType type) {
        if (type == QuizType.GUI) {
            return new GUIQuiz("Default GUI Quiz", new ArrayList<>()); // Provide default values
        } else {
            return new TerminalQuiz("Default Terminal Quiz", new ArrayList<>()); // Provide default values
        }
    }
}

```

```java
package il.ac.hit.quizzy;

import java.util.ArrayList;
import java.util.List;

public class QuizQuestion implements IQuizQuestion {
    private String title;
    private String question;
    private List<IQuizAnswer> answers = new ArrayList<>();

    private QuizQuestion() {
        // Private constructor to prevent direct instantiation
    }

    @Override
    public String getTitle() {
        return title;
    }

    @Override
    public String getQuestion() {
        return question;
    }

    @Override
    public List<IQuizAnswer> getAnswers() {
        return answers;
    }

    // Implement other methods

    // Implement the Builder class
    public static class Builder implements IQuizQuestionBuilder {
        private String title;
        private String question;
        private List<IQuizAnswer> answers = new ArrayList<>();
```

```
37
38          @Override
39          public IQuizQuestionBuilder setTitle(String
     text) {
40              this.title = text;
41              return this;
42          }
43
44          @Override
45          public IQuizQuestionBuilder setQuestion(
   String text) {
46              this.question = text;
47              return this;
48          }
49
50          @Override
51          public IQuizQuestionBuilder addAnswer(
   String text, boolean correct) {
52              IQuizAnswer answer = new QuizAnswer(
   text, correct);
53              answers.add(answer);
54              return this;
55          }
56
57          @Override
58          public IQuizQuestion create() {
59              QuizQuestion question = new
   QuizQuestion();
60              question.title = this.title;
61              question.question = this.question;
62              question.answers = this.answers;
63              return question;
64          }
65      }
66 }
67
```

```java
package il.ac.hit.quizzy;

import java.util.List;
import java.util.Scanner;

public class TerminalQuiz implements IQuiz {
    private String name;
    private List<IQuizQuestion> questions;

    public TerminalQuiz(String name, List<IQuizQuestion> questions) {
        this.name = name;
        this.questions = questions;
    }

    @Override
    public void start() {
        Scanner scanner = new Scanner(System.in);
        int score = 0;

        System.out.println("Welcome to " + name);

        // Iterate through questions and display them in the terminal
        for (IQuizQuestion question : questions) {
            System.out.println(question.getQuestion());

            // Display answer choices
            List<IQuizAnswer> answers = question.getAnswers();
            for (int i = 0; i < answers.size(); i++) {
                System.out.println((i + 1) + ". " + answers.get(i).getText());
            }

            // Get user's answer
            System.out.print("Enter the number of your answer: ");
            int userAnswer = scanner.nextInt();
```

```
35
36              // Check if the user's answer is
   correct
37              if (userAnswer >= 1 && userAnswer <=
   answers.size() && answers.get(userAnswer - 1).
   isCorrect()) {
38                  System.out.println("Correct!");
39                  score++;
40              } else {
41                  System.out.println("Incorrect.");
42              }
43          }
44
45          // Display the final score
46          System.out.println("Quiz completed! Your
   score: " + score + "/" + questions.size());
47      }
48
49      @Override
50      public void setName(String text) {
51          this.name = text;
52      }
53
54      @Override
55      public String getName() {
56          return name;
57      }
58
59      @Override
60      public void addQuestion(IQuizQuestion question
   ) {
61          questions.add(question);
62      }
63
64      @Override
65      public List<IQuizQuestion> getQuestions() {
66          return questions;
67      }
68 }
69
```

```java
package il.ac.hit.quizzy;

public interface IQuizFilesDAO {
    void saveQuizToFile(IQuiz quiz, String fileName) throws QuizException;
    IQuiz loadQuizFromFile(String fileName) throws QuizException;
}

```

```java
package il.ac.hit.quizzy;

import java.util.List;

public interface IQuizQuestion {
    String getTitle();

    String getQuestion();

    List<IQuizAnswer> getAnswers();

}

```

File - /Users/saarazoulay/Desktop/Quizzy/src/il/ac/hit/quizzy/QuizException.java

```java
package il.ac.hit.quizzy;

public class QuizException extends Exception {
    public QuizException(String message) {
        super(message);
    }
}

```

```
1 package il.ac.hit.quizzy;
2
3 public interface IQuizQuestionBuilder {
4     IQuizQuestionBuilder setTitle(String text);
5     IQuizQuestionBuilder setQuestion(String text);
6     IQuizQuestionBuilder addAnswer(String text,
  boolean correct);
7     IQuizQuestion create();
8 }
9
```

```java
package il.ac.hit.quizzy;

import java.io.*;
import java.util.List;

public class SimpleCSVQuizFilesDAO implements IQuizFilesDAO {
    private static SimpleCSVQuizFilesDAO instance;

    // Implement the Singleton pattern
    public static synchronized IQuizFilesDAO getInstance() {
        if (instance == null) {
            instance = new SimpleCSVQuizFilesDAO();
        }
        return instance;
    }

    @Override
    public void saveQuizToFile(IQuiz quiz, String fileName) throws QuizException {
        try (FileWriter writer = new FileWriter(fileName)) {
            List<IQuizQuestion> questions = quiz.getQuestions();
            for (IQuizQuestion question : questions) {
                // Write question and answers to the file in your custom format
                writer.write(questionToString(question) + "\n");
            }
        } catch (IOException e) {
            throw new QuizException("Error saving quiz to file: " + e.getMessage());
        }
    }

    @Override
    public IQuiz loadQuizFromFile(String fileName) throws QuizException {
```

```
32          try (BufferedReader reader = new
   BufferedReader(new FileReader(fileName))) {
33              IQuiz quiz = new Quiz();
34              String line;
35              while ((line = reader.readLine()) !=
   null) {
36                  // Parse the line to create
   questions and answers
37                  IQuizQuestion question =
   stringToQuestion(line);
38                  if (question != null) {
39                      quiz.addQuestion(question);
40                  }
41              }
42              return quiz;
43          } catch (IOException e) {
44              throw new QuizException("Error loading
   quiz from file: " + e.getMessage());
45          }
46      }
47
48      // Helper method to convert a question to a
   string
49      private String questionToString(IQuizQuestion
   question) {
50          StringBuilder builder = new StringBuilder
   ();
51          builder.append(question.getQuestion()).
   append(";");
52          List<IQuizAnswer> answers = question.
   getAnswers();
53          for (int i = 0; i < answers.size(); i++) {
54              IQuizAnswer answer = answers.get(i);
55              builder.append(answer.getText());
56              if (answer.isCorrect()) {
57                  builder.append("*"); // Use '*' to
   mark correct answers
58              }
59              if (i < answers.size() - 1) {
60                  builder.append(",");
61              }
```

```
62          }
63          return builder.toString();
64      }
65
66      // Helper method to convert a string to a
    question
67      private IQuizQuestion stringToQuestion(String
    line) {
68          String[] parts = line.split(";");
69          if (parts.length >= 2) {
70              String questionText = parts[0];
71              String[] answerStrings = parts[1].
    split(",");
72              IQuizQuestionBuilder builder = new
    QuizQuestion.Builder()
73                      .setQuestion(questionText);
74              for (String answerString :
    answerStrings) {
75                  boolean isCorrect = answerString.
    endsWith("*");
76                  if (isCorrect) {
77                      answerString = answerString.
    substring(0, answerString.length() - 1);
78                  }
79                  builder.addAnswer(answerString,
    isCorrect);
80              }
81              return builder.create();
82          }
83          return null;
84      }
85 }
86
```

```java
package il.ac.hit.quizzy;

import org.junit.Test;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import static org.junit.Assert.*;

public class SimpleCSVQuizFilesDAOTest {

    @Test
    public void testSaveAndLoadQuizToFile() {
        // Create a mock quiz with some questions
        IQuiz quiz = new Quiz();
        IQuizQuestion question1 = new QuizQuestion.Builder()
                .setQuestion("Question 1")
                .addAnswer("Answer 1", true)
                .addAnswer("Answer 2", false)
                .create();
        // Add more questions as needed
        quiz.addQuestion(question1);
        // Add more questions as needed

        // Define a temporary file name for testing
        String fileName = "testQuiz.data";

        // Use the SimpleCSVQuizFilesDAO to save the quiz to a file
        try {
            SimpleCSVQuizFilesDAO dao = (SimpleCSVQuizFilesDAO) SimpleCSVQuizFilesDAO.getInstance();
            dao.saveQuizToFile(quiz, fileName);

            // Check if the file was created
            File file = new File(fileName);
            assertTrue(file.exists());

            // Use FileReader and BufferedReader to
```

```
37  read the saved file
38              BufferedReader reader = new
   BufferedReader(new FileReader(file));
39              String line;
40              StringBuilder fileContent = new
   StringBuilder();
41              while ((line = reader.readLine()) !=
   null) {
42                  fileContent.append(line).append("\n
   ");
43              }
44              reader.close();
45
46              // Define the expected content based on
    your quiz format
47              String expectedContent = "Question 1;
   Answer 1*,Answer 2\n"; // Adjust this based on your
    format
48
49              // Use assertEquals to verify the
   expected and actual content
50              assertEquals(expectedContent,
   fileContent.toString());
51
52              // Clean up: delete the test file
53              file.delete();
54              assertFalse(file.exists());
55          } catch (QuizException | IOException e) {
56              // Handle exceptions if necessary
57              fail("Exception thrown: " + e.
   getMessage());
58          }
59      }
60 }
61
```