# Interim Assessment 4

EXCEPT FOR THE SECTIONS MARKED “ONLY REQUIRED FOR THE ASSIGNMENT”, Implement the next release of your term project. Implement the next release of your term project. You will incorporate the saving and retrieval of objects, the use of lambdas, and the use of streams. You can substitute the use of JavaFX for one of these.

The same instructions as in Assignment 2 applies to this completed Word document, the gray text, the 5 page limit, appendices, JUnit tests, and a ReadMe file.

## 1.1 SUMMARY DESCRIPTION

One- or two-paragraph overall description of your proposed term project. Color red the parts changed from Assignment 2.

This project is an educational tool, titled MusicTrivia, for those looking to test and improve their music theory knowledge. The application, a graphical user interface (GUI), will randomly generate questions from among several question templates ranging from simple to complex with random, appropriate values inserted at key points in the templates so that the user gets practice manipulating the questions with many different value permutations. Some of the concepts tested include major and minor scales, intervals, scale degree names, alto and tenor clefs, and diatonic chords. After the user has selected his or her answer from among multiple options, the application will offer feedback on the correctness of his or her answer.

In addition, the application will keep track of the user’s progress over time on each category of question so that the user can track his or her progress in specific topic areas.

## 1.2 ADDITIONAL REQUIREMENTS (FEATURES) IMPLEMENTED IN THIS RELEASE

Title and one or two sentences per requirement. Don’t repeat requirements implemented for prior assignments unless they are necessary to provide context—in which case, make it clear which are new vs. old.

### 1.2.1 Simple Clef Note Identification Questions (NEW)

Users will be asked to identify the notes located at various locations on the Alto and Tenor clef staves.

### 1.2.2 Multiple Choice Answers (NEW)

Users will be presented with four multiple choice options from which to choose the correct answer.

### 1.2.3 Graphical User Interface (NEW)

Users will have a graphical user interface (GUI) to interact with the application.

## (ONLY REQUIRED FOR THE ASSIGNMENT) 1.3 I/O SUPPORTING THE REQUIREMENTS LISTED ABOVE

Provide an example of input / output showing the new features of your application.

<Your response replaces this>

### Input

### Input / Output

### Output File

## (ONLY REQUIRED FOR THE ASSIGNMENT) 1.4 YOUR DIRECTORY

Show a screenshot of your directory. Include your “.dat” files (where objects are written—more on this later). This should include JUnit tests—class-by-class, and method-by-method, except for trivial ones.

<Your response replaces this>

## 1.5 DESIGN

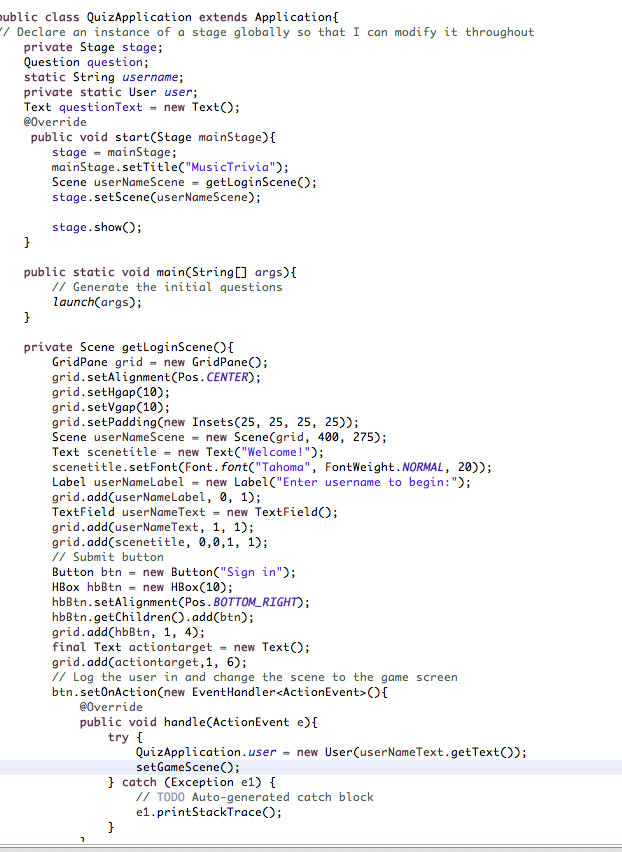
### 1.5.1 Class Model, Use Case, and Sequence Diagram

## Supply a main use case, the class model, and the sequence diagram corresponding to the use case. These should be consistent. Indicate in red your class model where you applied object read, object write, streams and lambdas.

### 

### The main use case is the user will be prompted to enter his/her username. Questions are then dynamically generated and the user answers questions and requests new questions. If an answer is selected by the user, feedback is given to the user about the answer. When the user “quits” the program, the User’s final total scores are displayed to the screen.

### 1.5.2 Code showing object read and write

Note: As per the instructions, I am choosing to substitute a JavaFX GUI for this requirement. Because of this, I am including the code for the JavaFX GUI here. Due to the relatively large amount of code, I am only including what could be caught in one screen shot. I believe this should give an idea of the code being written.

### (ONLY REQUIRED FOR THE ASSIGNMENT) 1.5.3 Code Showing *stream*() and Lambdas

<Your response replaces this>

## (ONLY REQUIRED FOR THE ASSIGNMENT) 1.7 YOUR CODE

Unless your facilitator arranges another method, copy your Eclipse project to your file system, zip it, and attach it. Please contact your facilitator in advance if you want to request an exception.

<Your response here>