

## IS 247 – Computer Programming II

### Final Exam Programming Exercise

#### Exercise Detail

For your final exam exercise you will create a quiz application that asks the user 5 multiple choice questions, prompts for an answer, then advises the user if their answer is correct or incorrect. If the user provides the wrong answer the application should provide the user with the correct answer. After the user completes the 5 questions the application should provide the user with a summary of their results including the number of correct and incorrect answers as well as their score (percent).

I am purposely not providing an example of the expected output. Be creative in how you present the questions, answers, and summary to the user but make sure that each aspect of your output is clean and easily readable.

#### The following items should be submitted:

1. Java file containing your solution named %LastName%\_%FirstName%\_Final.java (example Rodgers\_Tim\_Final.java). The class in your .java file should be named appropriately so that the file will compile and run as submitted.
2. Text file named %LastName%\_%FirstName%\_Questions.java (example Rodgers\_Tim\_Questions.txt) containing the questions for your solution.

#### Requirements:

- Your application must contain and use at least two custom methods in addition to the main method.
  - Quiz Questions
    - Quiz questions and answers must be read from a file named quiz.txt that is placed in the same folder as the java file being executed.
    - Questions should be multiple choice with the number of choices presented to the user for each question ranging from 2 to 5.
    - The questions.txt file should contain 10 or more question & answer combinations. The 5 questions asked should be randomly selected from the questions in this file.
    - You can choose to structure the contents of your question.txt file in any way you wish but keep in mind that it should be easy to add additional questions. (It is not part of this exercise, but you should consider how the file should be structured so that questions can be programmatically added and removed.)
    - The questions can be anything you would like (ie. state capitals); your application should be built in such a way that the subject of the questions is irrelevant.
- Here is a sample question:

What is the state capital of Maryland:

- A. Annapolis
- B. Baltimore
- C. Frederick
- D. Ocean City