Program Assignment 6

(25 points)

Due: Tuesday, November 13, 2018
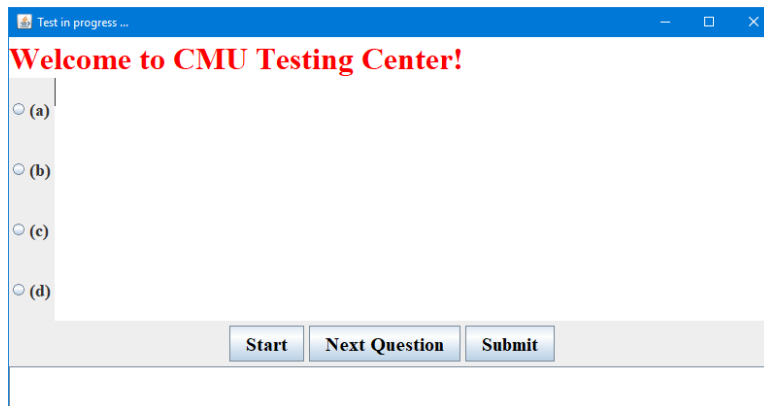
Purpose

This assignment to design a GUI using event model with clear object-oriented design.

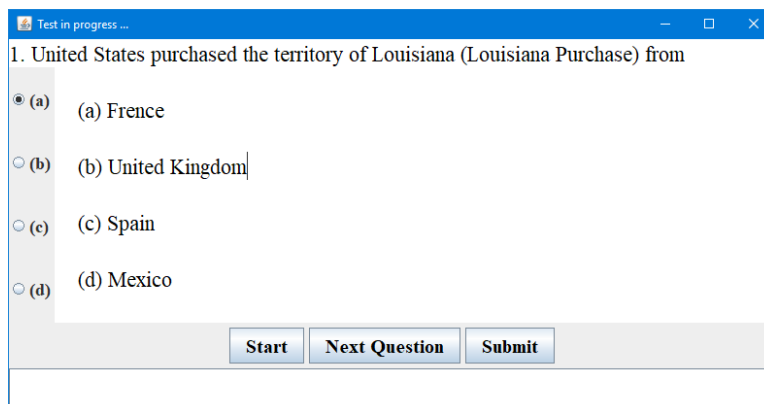## Graphic User Interface (GUI) of Testing System

Design and implement a testing system in Java. Write code using awt and swing in Java, rather than automatically generated code using Windowbuilder.

The GUI frame should have a text area for the display of the question statement, a group of radio buttons for the choices, a text area for display of the choices for the answers, three buttons ("Start", "Next Question", "Submit") for the user to act, and a text field to show the final score.

When your program starts, the GUI may look like this:

After the "Start" button is clicked, a question from the question bank is selected and displayed in the GUI:

Data – Question Bank

The testing system has a question bank of n multiple-choice questions stored in the file `questionBank.txt`
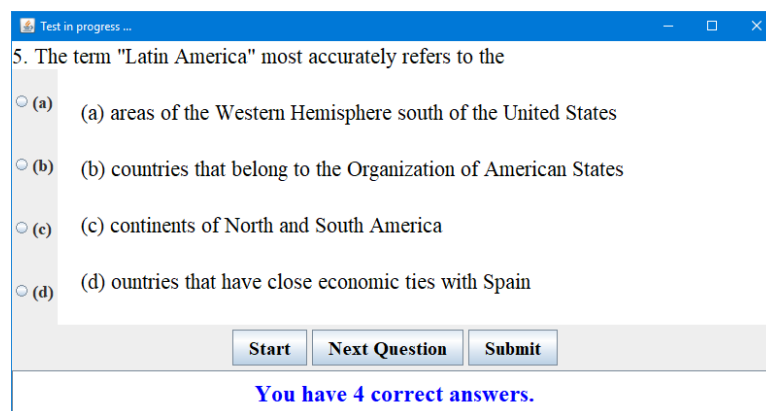
Each question has 6 lines in the file with this format:

Line 1	question number, tab, the question statement
Line 2–5	choice letter, tab, answer
Line 6	letter of the correct answer

Requirements

Five classes are required:

1. `Question`. A question should have a question statement, 4 choices labeled (a), (b), (c), (d), and a correct answer. It also provide appropriate access methods.
2. `QuestionBank`. The constructor of this method creates a list of questions from the data file. It also contains a method to randomly select m different questions from the bank of n questions $(5 \leq m \leq n)$ for a test.
3. A frame for the GUI. Use `BorderLayout`, similar to the figure shown above. Note: The choices, which is a `JTextArea`, should be displayed at the center.
4. An event handler that handles the button-click events. The main tasks are:
   (a) Start button: Get the first question and display in the frame.
   (b) Radio button: Figure out if the answer is correct or not and record the result.
   (c) Next button: Get the next question and display in the frame. Note: cannot go beyond the last question.
   (d) Submit button: Calculate the total number of correct answers and display in the text field, like this:

5. An application (with `main`) to create a `QuestionBank` object to initialize the questions from the data file, a GUI frame, an event handler, register the event handler with the buttons and radio buttons in the frame, and show the frame.

What to Submit

Submit your well-documented source program (`.java` files) on Blackboard.