

Programming Assignment #4

Trivia Quiz

CS 2308.256 Spring 2017

Instructor: Jill Seaman

Due: Tuesday, 3/21/2017: upload electronic copy by 9:00am.

Problem: Write a C++ program that will output multiple choice trivia questions.

Part I:

Design and implement a class called `Question`. The `Question` class contains the following information:

- a **stem** (the text of the trivia question),
- an array of 4 multiple choice **answers** (the text of each possible answer)
- the letter of the correct answer (A, B, C, or D), called the **key**.

This class will be used to represent trivia questions in a trivia game. The following operations should be available for `Question` objects (use the supplied names!).

- Construct a `Question` with no values (use empty strings for stem and answers and 'X' for the key).
- Construct a `Question` given its **3 components**.
- `setStem`: Set the **stem** question.
- `getStem`: Return the **stem** question.
- `setAnswers`: Set the **4 answers** given an array. The answers will NOT include the letter (A, B, C, or D).
- `getAnswer(i)`: Return the single answer at index `i`.
- `setKey`: Set the key letter.
- `getKey`: Return the key letter.
- `display`: Output the stem and answers (but not the key), each on a separate line. Output the letter A. before the first answer, B. before the second answer, and so on (you can use `char('A'+i)` to compute the letter for the question at index `i`). Put a period after each letter!

Save the class declaration in `Question.h` and save the member function definitions in `Question.cpp` (do not inline the member function definitions).

Use the driver on the class website: `QuestionTester.cpp` to test your class. You may modify this file to do more testing if you like.

Part II:

Design and implement a driver `QuizDriver.cpp`. This driver should create an array of 5 `Question` objects (initialized using the data provided in the file `questions.txt` on the class website). Then it should output the five questions (using the `display` member function). It should number the questions 1 to 5, with a dot and space after each number. It should also output a blank line after each question. Then it should output the key to the quiz (labeled "Answers"), all on one line with spaces between the keys.

Sample output:

1. What name is given to half of a Byte (4 bits)?

- A. Nibble
- B. Ort
- C. Scrap
- D. Snippet

2. Which country is home to the Kangaroo?

- A. China
- B. India
- C. Mexico
- D. Australia

3. What do you use to measure an angle?

- A. Compass
- B. Protractor
- C. Ruler
- D. T-Square

4. The Great Sphinx has the head of a human and the body of a what?

- A. Camel
- B. Eagle
- C. Lion
- D. Alligator

5. What is the flat rubber disc used in a game of ice hockey?

- A. Birdie
- B. Puck
- C. Dart
- D. Tile

Answers:

A D B C B

Create a **makefile** that will build your Trivia Quiz program!

NOTES:

- This program must be done in a **Linux or Unix** environment, using a command line compiler like g++. Do not use codeblocks, eclipse, or Xcode to compile.
- Your program **must compile** and run, otherwise you will receive a score of 0. Note: your Question.h and Question.cpp files must compile with the provided QuestionTester.cpp file, without changing it!
- Your program should NOT read the questions in from a file. You should copy the code from questions.txt into your driver. This is provided for your convenience.
- Put a header comment at the top of each source file (not the makefile)!
- The class declaration does not need comments (except for the header file). The function definitions in the *.cpp files DO need function header comments!!
- Turn in four files: Question.h, Question.cpp, QuizDriver.cpp, makefile. See below for instructions on how to combine them into one zip file.

Logistics:

Since there are multiple files for this assignment, you need to combine them into one file before submitting them. You should use the zip utility from the Linux/Unix command line:

```
[...]$zip assign4_XXXXX.zip Question.h Question.cpp  
QuizDriver.cpp makefile
```

This combines the 4 files into one zip file, **assign4_XXXXX.zip** (where XXXXX is your NetID). Then you should submit only assign4_XXXXX.zip.

There are two steps to the turn-in process:

1. Submit an electronic copy using the Assignments tool on the TRACS website for this class.
2. Submit a printout of the source files only (not the makefile) at the beginning of class, the day after the assignment is due. Please **print your name on top of the front page**, and staple if there is more than one page.

See the assignment turn-in policy on the course website (cs.txstate.edu/~js236/cs2308) for more details.