

CS 172 – Homework 1

Purpose:

After completing this assignment, you will have designed and implemented a class.

Description

There are two parts to the programming assignment.

1. Design and create a **Question** class according to the class specifications below. The **Question** class will hold data for a trivia question.
2. Write a script (on a separate file) that simulates a simple trivia game for two players.

Specification for Question class:

A Question object has a question or prompt and four possible answers, but only one of the answers is correct – so basically, we are simulating a multiple-choice question.

Attributes

- A trivia question or prompt
- Possible answer 1
- Possible answer 2
- Possible answer 3.
- Possible answer 4
- The number of the correct answer (either 1, 2, 3, or 4)

Note: you could also label the possible answers with letters (e.g.: a, b, c, and d). Then the correct answer would be a value such as a, b, c, or d.

Methods

- `__init__` method
- Getter methods (accessors/inspectors)
- Setter methods (mutators)
- A method that returns a string with the prompt and the possible answers. This method will be used in a script to display the trivia questions to the players. Make sure to format the string accordingly.

Script

Your program will simulate a simple trivia game for two players who will take turns in answering questions. Each time a player gives the correct answer to a question, the player earns a point. After all the questions have been answered, the program should display the number of points earned by each player and who the winner is, or indicate that there is a tie if that is the case.

Use the **Question** class to create at least 10 trivia questions on the topic of your choice. The **Question** objects you create must be stored in a list. If you need inspiration to create trivia questions, you may want to take a look at this site that has fan quizzes for many subjects (books, movies, TV shows, sports, etc.):

<https://www.allthetests.com/fan-quizzes-tests-Fantests.php>

NOTE: Please keep in mind that you are expected to write a good quality, well formatted program. That means:

- User input must be validated and your program gracefully handle invalid inputs.
- Repetitive code (code that appears in multiple places in the main script) should be written as a function.
- Your program must use good style, including proper identifier names, useful comments, and proper use of indentation and whitespace.
- Your program should also have an appropriate user interface so that anyone one using the program knows what to do and what to expect.

Sample program run

```
Welcome to the Python intro programming quiz
-----

Player 1 here is your question:
Which of the following is a valid variable name?
1. 3rdQuarter
2. quarter#3
3. quarter_3
4. quarter-3

Enter your answer: 0
Error: your answer has to be a value between 1 and 4. Try again.
Enter your answer: 1
That is incorrect. Better luck with the next question.

Player 2 here is your question:
The rules that must be followed when writing a program are called:
1. syntax
2. operators
3. punctuation
4. keywords

Enter your answer: 1
Excellent! You score!

And the final scores are:
Player 1: 0
Player 2: 1
Player 2 wins!
```

Grading

| Criteria | Points |
|---|------------|
| Question class: <code>__init__</code> method | 10 |
| Question class setters | 10 |
| Question class getters | 10 |
| Question class string method | 10 |
| Main script: user interface is easy to understand | 10 |
| Main script: properly creates at least 10 Question objects | 10 |
| Main script: Question objects are stored in a list | 10 |
| Main script: validates user's input | 10 |
| Main script: all the required parts are there: players taking turns, scoring and displaying the results | 10 |
| Code follows good style guidelines and separate repetitive code into functions | 10 |
| Total possible points | 100 |

NOTE: If you code has any runtime errors a 50-point deduction will be taken. Only portions of the code that execute without errors will be graded. If your script cannot run at all, you will receive 0 points.

How to Submit your assignment:

- Assignments must be submitted via Blackboard Learn.
 - Please note that assignments submitted via email will not be accepted.
 - Late assignments will not be accepted. Your work must be uploaded and submitted by 11:59 PM on the date it's due.
- For this assignment, you must submit a single zip (such as HW1.zip) file that contains:
 - `question.py` – file that contains the `Question` class.
 - `main.py` - your main script

Academic Honesty

You must be the **sole original author** of the **entire solution** you submit. You must compose all program and written material yourself. All material taken from outside sources (e.g. textbooks, in class examples, labs, etc.) must be appropriately cited.