Contents (../index.html) / Assignment #1: Candidate Testing

Assignment #1: Candidate Testing

OK, staff, we received many applications for our astronaut training program, and we need to do an initial evaluation of the candidates. Management needs you to create a quick quiz to help select the best candidates.

> **Note**
>
> The requirements below are what your END assignment will look like. This assignment is broken down so you can complete small pieces as you go. You need to move sequentially starting at Part 1. Please read the WHOLE assignment page before starting.

Requirements

1. Use functions to complete the following:
    1. Ask the candidate (user) to enter their name
    2. Use a loop to ask five questions, one at a time, to the candidate
    3. Collect the candidate's answers
    4. Check those answers for accuracy (case insensitive equality check)
    5. Calculate the candidate's overall percentage
    6. Determine if the candidate did well enough to enter our program (need >= 80% to pass)
    7. Display the results.

Take It Step by Step

When starting any project, it’s best to approach it as a series of smaller, testable parts. The goal is to get simple parts working first and then expand the code in a systematic way. The following is NOT the only way to complete this assignment, but it provides a framework for thinking through the project.

Get the Starter Code

Fork this repl.it (https://replit.com/@launchcode/candidate-testing-assignment-data-analysis).

Part 1: Minimum Viable Quiz

1.1 `candidate_name`

a. Ask for the candidate’s name. Look for **`TODO 1.1a`** in the starter code. On the line below this TODO comment, prompt the user for their name and store the value in a variable called **`candidate_name`**.
b. Look for **`TODO 1.1b`**. Underneath it, write a message to the console greeting the user using the name they just provided via your **`ask_for_name`** function.

1.2 Single Question Quiz

a. Ask the user to answer a single quiz question. Look for **`TODO 1.2a`**. Below the TODO comment, assign the variables called **`question`**, **`correct_answer`**, and **`candidate_answer`**.
    - **`question`** should be assigned the following string: **`"Who was the first American woman in space? "`**.

Tip

Note the trailing space at the end of this string is required.

   - **`correct_answer`** should be initialized to **`"Sally Ride"`**.
   - **`candidate_answer`** will initially be set to the empty string.

b. Find **`TODO 1.2b`**. Using your question variable, display the question and prompt the candidate for their answer. Store their response in one of the variables you defined just above.

c. Under **`TODO 1.2c`**, check the candidate's answer to see if it is correct. Provide basic feedback to the candidate, letting them know if their answer is correct or not.

Note

Make sure your small app works properly before moving on to part 2.

Part 2: Multiple Questions

Now that your small app is working, expand it to deal with multiple questions. This time, you only have one **`TODO`** item in the starter code. You will need to determine which lines need to be modified.

1. Define **`questions`** and **`correct_answers`** variables as lists. Use the table below to fill these lists.
2. Replace your code from **`TODO 1.2b`** with a loop that programmatically asks each question in the list and stores the user's responses.
3. Replace the basic feedback from **`TODO 1.2c`** with a template literal that displays each of the candidate's responses in addition to the corresponding correct answers.

Question	Answer
Who was the first American woman in space?	“Sally Ride”
True or false: 5 kilometer == 5000 meters?	“true”
(5 + 3)/2 * 10 = ?	“40”
Given the list `[8, 'Orbit', 'Trajectory', 45]`, what entry is at index 2?	“Trajectory”
What is the minimum crew size for the ISS?	“3”

⚠ **Warning**

Keep the questions and correct answers stored in this exact order. Feel free to copy-paste the questions and correct answers from the table to help avoid extraneous autograding errors.

Part 3: Grade the Quiz

Finally, calculate the candidate’s score and print the results. There are no `TODOs` in this section.

Your task here is to:

1. Compare the candidate answers with the correct answers,
2. Calculate the candidate’s score as a percentage,
3. Convey to the candidate if they have passed the quiz with an 80% or if they have failed.

Some tips:

1. Checking for the correct answer should be case insensitive (e.g. “Orbit” is the same as ”orbit”).
2. Somewhere below `TODO 1.2c` you should see a variable called `grade`. Use this to calculate the candidate’s score.
3. To calculate the candidate’s percentage, use the equation:

(Number of Correct Answers) / (Number of Quiz Questions) * 100

Example Output

The results output should include the candidate's name, the candidate's responses, the correct answers, the final percentage, and if the candidate passed the quiz.

```
Candidate Name: Can Twin
1) Who was the first American woman in space?
Your Answer: sally ride
Correct Answer: Sally Ride

2) True or false: 5000 meters = 5 kilometers.
Your Answer: false
Correct Answer: true

3) (5 + 3)/2 * 10 = ?
Your Answer: 45
Correct Answer: 40

4) Given the list [8, "Orbit", "Trajectory", 45], what entry is at index 2?
Your Answer: trajectory
Correct Answer: Trajectory

5) What is the minimum crew size for the ISS?
Your Answer: 10
Correct Answer: 3

>>> Overall Grade: 40% (2 of 5 responses correct) <<<
>>> Status: FAILED <<<
```

Note

The output will vary slightly based on the candidate's answers to each question.

Submitting Your Work

When your program meets the requirements and the output looks roughly like the example above, you've finished your assignment and can submit the url to your repl.it in your Canvas assignment.

← Index (../genindex.html)

Assignment #2: Call Center Cleanup → (call-center-cleanup.html)

Back to top