Excellence

NZQA Intended for teacher use only

**Purpose of Quiz**
The purpose of my quiz program is to test and extend Year 11 students' knowledge of Pandora's Box from Greek Mythology.

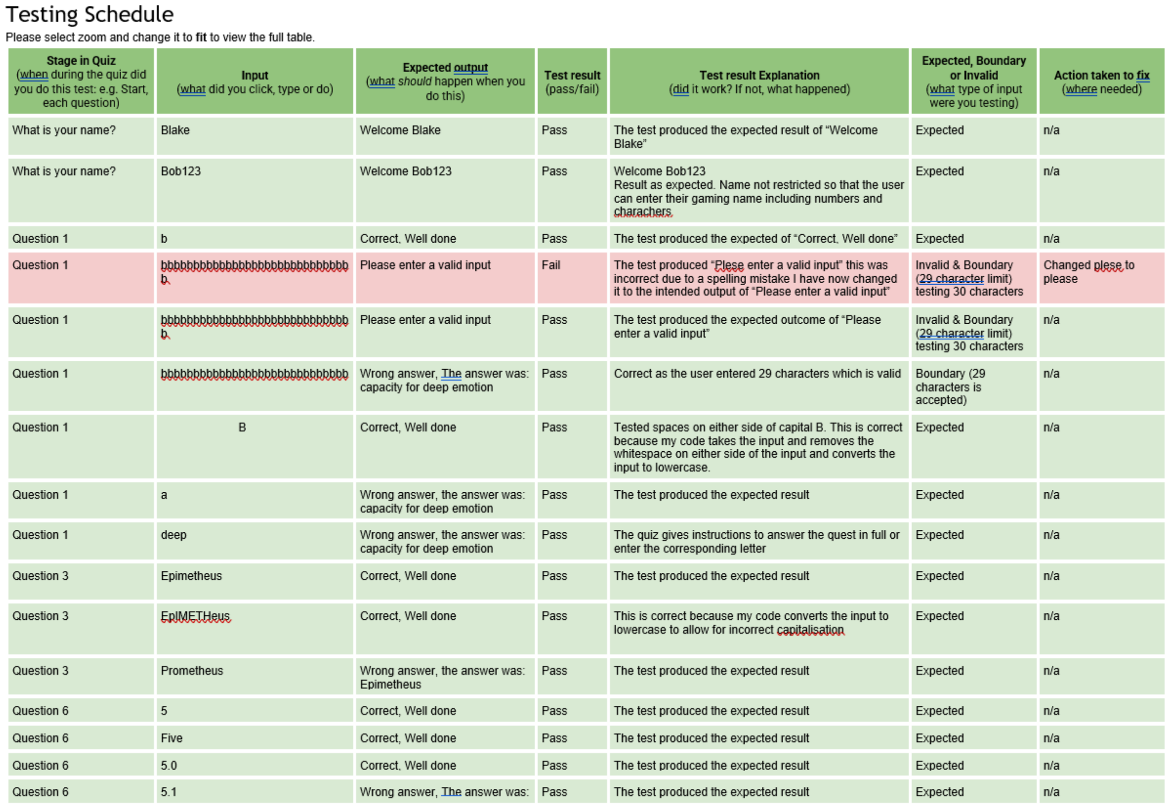
**Style of question** (e.g. multiple choice, short answer)
A mixture of multiple choice and short answer questions

**Example question and answer**
What did Aphrodite give to Pandora?
(a) Mastery over language
(b) Capacity for deep emotion
(c) Fine craftmanship and attention to detail
(d) The trait of curiosity

**Scoring system**
1 point per question answered correctly.

**When quiz ends**
The quiz ends when the user has completed all the questions.

**Boundary conditions I could test**
If a user doesn't enter an answer, prompt them for an input. If the user enters more than 30 characters prompt them for a correct answer.

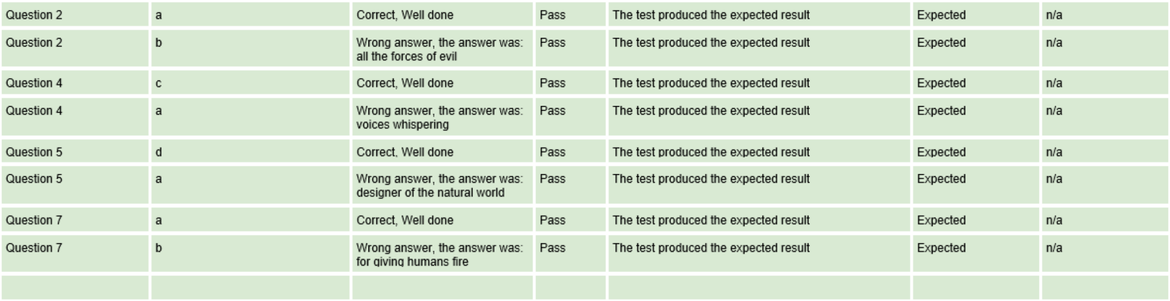


Testing Schedule

Please select zoom and change it to **fit** to view the full table.

| **Stage in Quiz**<br>(when during the quiz did you do this test: e.g. Start, each question) | **Input**<br>(what did you click, type or do) | **Expected output**<br>(what *should* happen when you do this) | **Test result**<br>(pass/fail) | **Test result Explanation**<br>(did it work? If not, what happened) | **Expected, Boundary or Invalid**<br>(what type of input were you testing) | **Action taken to fix**<br>(where needed) |
|---|---|---|---|---|---|---|
| What is your name? | Blake | Welcome Blake | Pass | The test produced the expected result of "Welcome Blake" | Expected | n/a |
| What is your name? | Bob123 | Welcome Bob123 | Pass | Welcome Bob123<br>Result as expected. Name not restricted so that the user can enter their gaming name including numbers and charachers | Expected | n/a |
| Question 1 | b | Correct, Well done | Pass | The test produced the expected of "Correct, Well done" | Expected | n/a |
| Question 1 | bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb | Please enter a valid input | Fail | The test produced "Plese enter a valid input" this was incorrect due to a spelling mistake I have now changed it to the intended output of "Please enter a valid input" | Invalid & Boundary (29 character limit) testing 30 characters | Changed plese to please |
| Question 1 | bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb | Please enter a valid input | Pass | The test produced the expected outcome of "Please enter a valid input" | Invalid & Boundary (29 character limit) testing 30 characters | n/a |
| Question 1 | bbbbbbbbbbbbbbbbbbbbbbbbbbbbb | Wrong answer, The answer was: capacity for deep emotion | Pass | Correct as the user entered 29 characters which is valid | Boundary (29 characters is accepted) | n/a |
| Question 1 | B | Correct, Well done | Pass | Tested spaces on either side of capital B. This is correct because my code takes the input and removes the whitespace on either side of the input and converts the input to lowercase. | Expected | n/a |
| Question 1 | a | Wrong answer, the answer was: capacity for deep emotion | Pass | The test produced the expected result | Expected | n/a |
| Question 1 | deep | Wrong answer, the answer was: capacity for deep emotion | Pass | The quiz gives instructions to answer the quest in full or enter the corresponding letter | Expected | n/a |
| Question 3 | Epimetheus | Correct, Well done | Pass | The test produced the expected result | Expected | n/a |
| Question 3 | EpIMETHeus | Correct, Well done | Pass | This is correct because my code converts the input to lowercase to allow for incorrect capitalisation | Expected | n/a |
| Question 3 | Prometheus | Wrong answer, the answer was: Epimetheus | Pass | The test produced the expected result | Expected | n/a |
| Question 6 | 5 | Correct, Well done | Pass | The test produced the expected result | Expected | n/a |
| Question 6 | Five | Correct, Well done | Pass | The test produced the expected result | Expected | n/a |
| Question 6 | 5.0 | Correct, Well done | Pass | The test produced the expected result | Expected | n/a |
| Question 6 | 5.1 | Wrong answer, The answer was: | Pass | The test produced the expected result | Expected | n/a |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | 5 | | | | |
| Score | Score is 0 out of 7 | Well done you have completed the quiz *username* you got x out of 7 | Pass | The test produced the expected result | Expected | n/a |
| Score | Score is calculated to 0% | You do not know Pandora's Box very well. Do you want to watch the story? | Fail | Traceback error message | Boundary | Miss spelling of variable name in percentage calculation. Corrected error |
| Score | Score is calculated to 0% | You do not know Pandora's Box very well. Do you want to watch the story? | Pass | The test produced the expected result. Boundary testing to check the correct message is returned based on the percentage score | Boundary | n/a |
| Score | Score is 1 out of 7 | Well done you have completed the quiz *username* you got 1 out of 7 | Pass | The test produced the expected result | Expected | n/a |
| Score | Score is calculated to 14% | You do not know Pandora's Box very well. Do you want to watch the story? | Pass | The test produced the expected result. Boundary testing to check the correct message is returned based on the percentage score | Boundary | n/a |
| Score | Score is 2 out of 7 | Well done you have completed the quiz *username* you got 2 out of 7 | Pass | The test produced the expected result | Expected | n/a |
| Score | Score is calculated to 29% | You do not know Pandora's Box very well. Do you want to watch the story? | Pass | The test produced the expected result. Boundary testing to check the correct message is returned based on the percentage score | Boundary | n/a |
| Score | Score is 3 out of 7 | Well done you have completed the quiz *username* you got 3 out of 7 | Pass | The test produced the expected result | Expected | n/a |
| Score | Score is calculated to 43% | You do not know Pandora's Box very well. Do you want to watch the story? | Pass | The test produced the expected result. Boundary testing to check the correct message is returned based on the percentage score | Boundary | n/a |
| Score | Score is 4 out of 7 | Well done you have completed the quiz *username* you got 4 out of 7 | Pass | The test produced the expected result | Expected | n/a |
| Score | Score is calculated to 57% | Well done you know Pandora's Box quite well | Fail | Incorrect message returned. Message for less than 50% returned. | Boundary | Updated user score to percentage. This makes sure that the message returned is based on the percentage score, not the total out of 7. |
| Score | Score is calculated to 57% | Well done you know Pandora's Box quite well | Pass | The test produced the expected result. Boundary testing to check the correct message is returned based on the percentage score | Boundary | n/a |
| Score | Score is 5 out of 7 | Well done you have completed the quiz *username* you got 5 out of 7 | Pass | The test produced the expected result | Expected | n/a |
| Score | Score is calculated to 71% | Well done you know Pandora's Box quite well | Pass | The test produced the expected result. Boundary testing to check the correct message is returned based on the percentage score | Boundary | n/a |
| Score | Score is 6 out of 7 | Well done you have completed the quiz *username* you got 6 out of 7 | Pass | The test produced the expected result | Expected | n/a |
| Score | Score is calculated to 85% | Well done you are a Pandora's Box expert | Pass | The test produced the expected result. Boundary testing to check the correct message is returned based on the percentage score | Boundary | n/a |
| Score | Score is 7 out of 7 | Well done you have completed the quiz *username* you got 7 out of 7 | Pass | The test produced the expected result | Expected | n/a |
| Score | Score is calculated to 100% | Well done you are a Pandora's Box expert | Pass | The test produced the expected result. Boundary testing to check the correct message is returned based on the percentage score | Boundary | n/a |
| Do you want to watch the video? | Yes | Video plays in web browser | Pass | Video plays | Expected | n/a |
| Do you want to watch the video? | No | Do you want to play again? | Pass | The test produced the expected result | Expected | n/a |
| Do you want to watch the video? | x | Invalid input please enter yes or no<br>Ask the question again | Fail | Moved on to the next question.<br>Do you want to play again? | Invalid | Add code to check if it is an accepted input |
| Do you want to watch the video? | x | Invalid input please enter yes or no<br>Ask the question again | Pass | The test produced the expected result | Invalid | n/a |
| End<br>Do you want to play again? | i | Invalid input | Fail | Game ends | Invalid | Add a condition to check for a valid input |
| End<br>Do you want to play again? | no | End game | Fail | Invalid input please enter yes or no<br>This is an incorrect message the game should have ended | Expected | Changed 'or' to 'and' in the if statement that checks for a valid input |
| End<br>Do you want to play again? | i | Invalid input please enter yes or no<br>Ask the question again | Pass | Print: Invalid input please enter yes or no<br>Ask question again | Invalid | n/a |
| End<br>Do you want to play again? | Yes | Game restarts | Pass | The test produced the expected result | Expected | n/a |
| End<br>Do you want to play again? | sure | Game restarts | Pass | The test produced the expected result because I have included a list of valid yes responses which includes sure. | Expected | n/a |
| Start | Start | Game should play | Fail | Quiz would not start | Expected | Error after changing variable names. Fixed by correcting variable name. |



| | | | | | | |
|---|---|---|---|---|---|---|
| Question 2 | a | Correct, Well done | Pass | The test produced the expected result | Expected | n/a |
| Question 2 | b | Wrong answer, the answer was: all the forces of evil | Pass | The test produced the expected result | Expected | n/a |
| Question 4 | c | Correct, Well done | Pass | The test produced the expected result | Expected | n/a |
| Question 4 | a | Wrong answer, the answer was: voices whispering | Pass | The test produced the expected result | Expected | n/a |
| Question 5 | d | Correct, Well done | Pass | The test produced the expected result | Expected | n/a |
| Question 5 | a | Wrong answer, the answer was: designer of the natural world | Pass | The test produced the expected result | Expected | n/a |
| Question 7 | a | Correct, Well done | Pass | The test produced the expected result | Expected | n/a |
| Question 7 | b | Wrong answer, the answer was: for giving humans fire | Pass | The test produced the expected result | Expected | n/a |
| | | | | | | |

```
#imports the web feature and allows use of the users default web browser
import webbrowser

#classes enable greater flexibility to add more questions easily in the future
#Sets up the class to store the players name and the players score
class Player:
        def __init__(self, name, score):
                self.name = name
                self.score = score
#Sets up class to store the questions and answers for the quiz. A class makes it easier to setup more questions and answers in the future.
#There are only 2 steps to follow to add extra questions and answers.
class Quiz:
        def __init__(self, question, answers):
                self.question = question
                self.answers = answers


#questions stored in a list for greater flexibility.
#Step one to add another question is to add the question to this list.
list_questions = [
        "What did Aphrodite give to Pandora?\n(a) Mastery over language\n(b) Capacity for deep emotion\n(c) Fine craftmanship and attention to detail\n(d) The trait of curiosity\n",
        "What was in Pandora's box?\n(a) All the forces of evil\n(b) A portel to hell\n(c) A titan\n(d) Nothing\n",
        "Who did Pandora fall in love with?\n",
        "What sound did Pandora hear from the box?\n(a) Music\n(b) Animals\n(c) Voices whispering\n(d) Laughing\n",
        "What was Epimetheus's job?\n(a) Builder\n(b) God of Fire\n(c) God of Water\n(d) Designer of the natural world\n",
        "How many Gods helped to create Pandora?\n",
        "Why was Prometheus eternally punished?\n(a) For giving humans fire\n(b) For creating humans\n(c) For falling in love with Pandora\n(d) For giving humans weapons\n"

]


#stores question and answer data in a list for greater flexibility for adding more questions and/or answers
#Step two to setting up new questions is copy and paste the bottom line of this list, update to the next number and put the required answer
list_questions_answers = [
        Quiz(list_questions[0], ["capacity for deep emotion", "b"]),
```

© NZQA 2023

```
    Quiz(list_questions[1], ["all the forces of evil", "a"]),
    Quiz(list_questions[2],  ["epimetheus"]),
    Quiz(list_questions[3], ["voices whispering", "c"]),
    Quiz(list_questions[4], ["designer of the natural world", "d"]),
    Quiz(list_questions[5], ["5", "5.0", "five"]), Quiz(list_questions[6], ["for giving humans fire",
    "a"])

]


#list of accepted answers for yes to allow for flexibility in users input
yes_parameters = ["yes", "y", "ok", "sure"]
#list of accepted answers for no to allow for flexibility in users input
no_parameters = ["no", "n", "no thanks"]
#define function so that this code can be called apon anywhere
def run_quiz_program(list_questions_answers):
    #Gets players name and set score to 0
    user = Player(input("What is your name?\n") ,0)
    print("Welcome", user.name, "\nYou must enter the corresponding letter for your chosen
      answer or type the full answer.\n")

    #Cycles the game through each question in the quiz until it reaches the end
    for Quiz in list_questions_answers:
        #using a loop to ensure user puts an input in instead of just clicking enter
        while True:
            #sets input to lowercase, removes whitespace to the left and right of text so that
              the user response will be correct when it has incorrect formating
            user_answer = input(Quiz.question).lower().strip()
            #If no answer is given or the answer has more than 30 characters the input is rejected
              and the user is asked for a valid input
            if len(user_answer) == 0 or len(user_answer) >= 30:
                print("Please enter a valid input\n")
                continue
            else:
                 break

        #add a point to user score if user answer is correct
        if user_answer in Quiz.answers:
            user.score += 1
            print("Correct, Well done\n")
        else:
            print("Wrong answer, The answer was:", Quiz.answers [0],"\n")

    #returns users score
    print("Well done you have completed the quiz", user.name, "you got", user.score, "out of",
      str(len(list_questions_answers)),"\n")

    #calculate score percentage. Return comment based on percentage score
    percentage = 100 * float(user.score)/float(str(len
      (list_questions_answers)))

    if percentage >= 80:
        print("Well done you are a Pandora's Box expert")
    elif percentage >= 50:
        print("Well done you know Pandora's Box quite well")
    else:
        #using a loop to ensure the player gives a valid yes/no answer
        while True:
            watch_video = input("You do not know Pandora's Box very well. Do you want to
              watch the story?\nYes/No\n").lower ().strip()
            #Check user response against list of possible yes and no answers. If answer is not valid
              ask the question again. This allows flexibility in the way the user answers yes or no
            if watch_video not in yes_parameters and watch_video not in no_parameters:
                print("Invalid input please enter yes or no")
                continue
            elif watch_video in yes_parameters:
                webbrowser.open_new("https://www.youtube.com/watch?
                  v=pMdJxVjZMRI")
                print("If the video does not start playing please check your browser")
                break
            else:
                break


    #using a loop to ensure the player gives a valid yes/no answer
    while True:
        play_again = input("Do you what to play again?\nYes/No
          \n").lower().strip()
        #Check user response against list of possible yes and no answers. If answer is not valid ask
          the question again. This allows flexibility in the way the user answers yes or no
        if play_again not in yes_parameters and play_again not in no_parameters:
            print("Invalid input please enter yes or no")
            continue
        #if the answer is in the yes list play agin
        elif play_again in yes_parameters:
            run_quiz_program(list_questions_answers)
            break
        else:
            print("Thank you for playing", user.name)
            break


#start the quiz
run_quiz_program(list_questions_answers)

```