National Certificate of Educational Achievement
TAUMATA MĀTAURANGA Ā-MOTU KUA TAEA

Exemplar for Internal Achievement Standard

Digital Technologies Level 1

This exemplar supports assessment against:

Achievement Standard 92004

Create a computer program

An annotated exemplar is a sample of student evidence, with a commentary, to explain key aspects of the standard. It assists teachers to make assessment judgements at the grade.

New Zealand Qualifications Authority

To support internal assessment

Grade: Achieved
For Achieved, the standard requires the student to create a computer program. This involves using a suitable programming language to construct a program that performs a specified task. The program needs to store at least two types of data in variables, take input, produce output; use sequence, selection and iteration control structures; and use data stored in a collection. The program must be tested and debugged to ensure it works on expected cases, and documented with comments. This student constructed a program in Python to quiz classmates on NRL. The program has data stored in integers and strings, and accepts input from users and outputs answers and scores. ‘If’ statements and ‘while’ loops meet the requirement for using control structures. The questions and answers are stored in arrays, showing the use of collections. Only one collection is required to meet the standard. The testing and debugging table demonstrates that the program worked on expected cases. The notes column meets the requirement for debugging and shows the changes made by the student. The program has been documented with comments throughout the code. For Merit, the standard requires boundary cases to be tested and debugged, and commenting should clarify code sections. For example, the program could have messages added for high, medium, and low scores, and additional testing carried out to check boundaries for the score. Commenting in the program could be amalgamated and simplified to clarify the purpose of code sections.

What is to be done?
I have to create a quiz about a pakiwaitara (stories) to share with my classmates. I have to choose a pakiwaitara that is relevant to me and collect information about it to make sure I understand it well. Based on that information, I need to write a program that presents a quiz that could be used by my classmates to test or extend their knowledge of my topic.

Who is it for?
The quiz I am creating will be for my classmates and my teacher.

Why is it to be done?
This is to be done because it is an assessment. It will also teach my classmates new things when they have finished the quiz and will hopefully be a great help to them.

Specifications and Requirements
- Ask a minimum of 5 questions.
- Keep and display a score.
- Store at least two types of data in variables
- Take input and produce output.
- Use conditionals (such as if and else) and loops.
- Use data stored in a collection (such as a list or arrays)

Testing and debugging

Test (enter)	Output expected	Correct?	Notes
Question: How many Grand Finals has your team won? Value = 7	User is prompted to give question input	YES	
Question: How many Grand Finals has your team won? Value = 25	User is prompted to give question input	YES	
Question: Please enter a valid number of Grand Finals from 8 to 21. Value = A letter	User sees error message and prompts to give valid value for the question	YES	I just needed to space out the words because you couldn’t read it properly as it was all close together.
Question: Who won the 2010 Grand Final? Value = C	User enters incorrect value and is told that the correct answer is D.	YES	

Question: This trivia is intended for teams with 8 to 21 Grand Finals. Value = Quit	User quits and the trivia stops.	NO	I checked the code and realised I didn't enter the correct coding for the quit button, so now, if they press quit, they will continue the quiz.
Question: All of them User presses enter button on keyboard without choosing an answer.	User moves on to the next question and answer is random.	No	
User enters a letter instead of a valid answer when asked for how many Grand Finals they have won	User is asked to enter a valid number.	Yes	

Ongoing Improvements
My ongoing improvements are that I tested my programme and searched for things I needed to fix. I saw that when I entered the correct or incorrect answer, I would get the expected outcome but could not read it properly because I did not space out the words. I also saw that when I wanted quit the game I would continue it so I had to fix that as well.

```python
#Import Easygui function so gui can be created

import easygui

Grand_Final_MIN = 8
Grand_Final_MAX = 21
Trivia_Grand_Final = 10
MAX_QUESTION_ATTEMPTS = 2

#Ask the user's NRL Team - String Variable

title = "Welcome to the NRL Trivia"
msg = "What is your NRL Team?"
NRL_Team = ""

#Ask the user how many Grand Finals they've won - Numeric Variable
while NRL_Team == "":
        NRL_Team = easygui.enterbox(msg, title, "")

title = "Welcome to the NRL Start Trivia"
msg = "How many Grand Finals has your team won?"

#Check the Grand Final criteria for playing the Trivia. Also checks if the player has entered an integer within a valid Grand Final range. The loop repeats until a valid integer is entered. - Iteration
Grand_Final = easygui.integerbox(msg, title, "")
while Grand_Final < Grand_Final_MIN or Grand_Final > Grand_Final_MAX:
        msg = "Please enter a valid amount of Grand Finals from " +  str (Grand_Final_MIN) + " to " + \
        str(Grand_Final_MAX) + " Grand Finals."
        Grand_Final = easygui.integerbox(msg, title, "")

#Checks whether the player falls within the Trivia Grand Final range
continue_game = "Continue"
if Grand_Final >= Trivia_Grand_Final:
        print (Grand_Final)
        msg = "This Trivia is intended for Teams with 8 to 21 Grand Final Championships."
        choices = ["Continue", "Quit"]
        continue_game = easygui.buttonbox(msg, title, choices=choices)
        print (continue_game)

#This is the gate to check whether the quiz should continue because either the user has earlier indicated they are under the Trivia Grand Final, or they want to continue even though they are older.
if continue_game == "Continue":
        title = "Welcome to the NRL Trivia"
        msg = "Hey "  + NRL_Team + "! Just before we start, the only rule is that you are not allowed to search up the answers. If you do not know the answer, just take a guess or try really hard to remember it. Anyways, enjoy the Trivia and may the best team win."
        ok_button = "Start"
        easygui.msgbox(msg, title, ok_button)
```

```

#Setup questions and answers for players - Data stored in List
questions_a = ["Who won the 2010 Grand Final?\n\nA: Storms\nB: Sydney Roosters\nC: Cowboys\nD: ST George Illawarra Dragons\n",
               "How many teams are there in the NRL?\n\nA: 15\nB: 14\nC: 16\nD: 17\n",
               "Who is the current hooker that plays for the Rabbitohs?\n\nA: Harry Grant\nB: Damien Cook\nC: Api Koroisau\nD: Brandon Smith\n",
               "Which person has played for 3 different teams?\n\nA: Josh Addo - Carr\nB: James Tedesco\nC: Brian To'o\nD: Latrell Mitchel\n",
               "Who is the Coach for the Parramatta Eels?\n\nA: Wayne Barrett\nB: Brad Arthur\nC: Ricky Stuart\nD: Anthony Griffin\n",
               "How many points is a try worth?\n\nA: 5\nB: 6\nC: 4\nD: 7\n",
               "How many meters on a full NRL field?\n\nA: 110\nB: 105\nC: 100\nD: 1000\n",
               "What happens if someone drops the ball?\n\nA: Drop Kick\nB: Foward Pass\nC: Knock On\nD: Double Dribble\n",
               "Who has scored the most points in one game?\n\nA: Trent Robbinson\nB: Dave Brown\nC: Nathan Cleary\nD: Josh Papali'i\n",
               "Which is the best NRL team?\n\nA: Rabbitohs\nB: Rabbitohs\nC: Rabbitohs\nD: All of the above\n"]

#Setup answers to the multiple questions - Data stored in List
answers_a=["D","C","B","A","B","C","C","C","B","A"]

#Set Question score to zero to start the Program with no score - Data stored in List
q_score=0

#Question 1 - Selection
player_trivia = easygui.buttonbox(questions_a[0],"Questions 1",choices= ["A","B","C","D"])
if player_trivia == answers_a[0]:
        easygui.msgbox("WOW, "  +  NRL_Team + "! "  +  " Good Job!")
        q_score = q_score + 1
else:
        q_response = easygui.msgbox("WOW, " + NRL_Team + " !  Guess your not winning this year.\nThe correct answer is " + answers_a[0])

#Question 2 - Selection
player_trivia = easygui.buttonbox(questions_a[1],"Questions 2",choices= ["A","B","C","D"])
if player_trivia == answers_a[1]:
        easygui.msgbox("Fantastic, "  +  NRL_Team + "! "  +  " Doing Great!")
        q_score = q_score + 1
else:
        q_response = easygui.msgbox("ERR ERRRRRRRR, " + NRL_Team + "!
        Wrong one.\nThe correct answer is " + answers_a[1])

#Question 3 - Selection
player_trivia = easygui.buttonbox(questions_a[2],"Questions 3",choices= ["A","B","C","D"])
if player_trivia == answers_a[2]:
        easygui.msgbox("Amazing, " + NRL_Team + "! " + " Keep it up!")
        q_score = q_score + 1
else:
        q_response = easygui.msgbox("Nope, " + NRL_Team + "!   Wrong again. \nThe correct answer is " + answers_a[2])
#Question 4 - Selection
player_trivia = easygui.buttonbox(questions_a[3],"Questions 4",choices= ["A","B","C","D"])
if player_trivia == answers_a[3]:
        easygui.msgbox("Outstanding, " + NRL_Team + "! " + " You're on a roll")
        q_score = q_score + 1
else:
        q_response = easygui.msgbox("Really?, " + NRL_Team + "!   That one was easy.\nThe correct answer is " + answers_a[3])

#Question 5 - Selection
player_trivia = easygui.buttonbox(questions_a[4],"Questions 5",choices= ["A","B","C","D"])
if player_trivia == answers_a[4]:
        easygui.msgbox("Impossible, " + NRL_Team + "! " + " Let's see if you make it to the finals")
        q_score = q_score + 1
else:
        q_response = easygui.msgbox("Come on, " + NRL_Team + "!   Unlucky. \nThe correct answer is " + answers_a[4])

#Question 6 - Selection
player_trivia = easygui.buttonbox(questions_a[5],"Questions 6",choices= ["A","B","C","D"])
if player_trivia == answers_a[5]:
        easygui.msgbox("That's Crazy, " + NRL_Team + "! " + " Almost there")
        q_score = q_score + 1
else:
        q_response = easygui.msgbox("Wrong, " + NRL_Team + "!   Guess you had a bad game.\nThe correct answer is " + answers_a[5])

#Question 7 - Selection
player_trivia = easygui.buttonbox(questions_a[6],"Questions 7",choices= ["A","B","C","D"])
if player_trivia == answers_a[6]:
        easygui.msgbox("ALL RIGHT, " + NRL_Team + "! " + " That was a fluke")
        q_score = q_score + 1
else:
        q_response = easygui.msgbox("NO WAY, " + NRL_Team + "!   Are you
serious.\nThe correct answer is " + answers_a[6])

#Question 8 - Selection
player_trivia = easygui.buttonbox(questions_a[7],"Questions 8",choices= ["A","B","C","D"])
if player_trivia == answers_a[7]:
        easygui.msgbox("Let's go, " + NRL_Team + "! " + " 2 more to go")
        q_score = q_score + 1
else:
        q_response = easygui.msgbox("Come on now, " + NRL_Team + "!   NO NO NO.\nThe correct answer is " + answers_a[7])

#Question 9 - Selection
player_trivia = easygui.buttonbox(questions_a[8],"Questions 9",choices= ["A","B","C","D"])
if player_trivia == answers_a[8]:
        easygui.msgbox("OK, " + NRL_Team + "! " + " That one was easy")
        q_score = q_score + 1
else:
        q_response = easygui.msgbox("WOW, " + NRL_Team + "!   Guess your not winning this year.\nThe correct answer is " + answers_a[8])

#Question 10 - Selection
player_trivia = easygui.buttonbox(questions_a[9],"Questions 10",choices=["A","B","C","D"])
if player_trivia == answers_a[9]:
        easygui.msgbox("Perfect pick, " + NRL_Team + "! " + " I always knew you were a Rabbitohs fan")
        q_score = q_score + 1
else:
        q_response = easygui.msgbox("Perfect pick, " + NRL_Team + "!   I always knew you were a Rabbitohs fan.")

#Tell the user the amount of Grand Finals they have won out of 10
easygui.msgbox(str(NRL_Team)   +   ",you have won   "   +   str(q_score)   +   " Grand Finals.\nYour score: " + str(q_score) + "/10","NRL Trivia")

#Displays message when player opts to quit the game or when all questions have been answered
title = "NRL Trivia"
msg = "Have a great rest of your season!"
button = "Close"
easygui.msgbox(msg, title, button)

```

Grade: Merit
For Merit, the student needs to create a well-structured computer program. This involves using succinct and descriptive variable names, documenting the program with comments that clarify the purpose of code sections, and testing and debugging the program to ensure it works on expected and boundary cases. This student has used succinct variable names that describe what the variable is used for. For example, **questions** is used to store the ‘array’ of questions. Comments have been written in the program at the top of code sections to describe their purpose. Testing has been carried out on both expected and boundary cases. For example, boundary tests of 2, 3 and 4 have been tested for the score messages. Debugging is evident from the two versions of testing and the changes that were made. For Excellence, the standard requires conditions and control structures to be used effectively. For example, the student could reduce the repeated code by using a loop or functions.

Merit

NZQA Intended for teacher use only

```
#MyQuiz v1.1
#
#This gets the player to answer questions in a quiz i have setted up.

#The score code is not here because the score would add on to the previouse games score, but
    now that is placed inside the loop it resets to 0 every time the player wants to play again.
PASS = 3
FLAWLESS = 5

#This list is the asortment of the questions for the quiz.
questions = ["\nA family reunion is a mass gathering of family members that have been apart in a
    period of time (True or False): \n",
                "\nThe largest recored family reunion is called the Lilly  Family Reunion, but do you know
                    how many attended? \n1 - 150
                      \n2 - 1,000 \n3 - 400 \n4 - 125 \n5 - 2,5000\nEnter answer here:",
            "\nIs it okay to attend another family's reunion?\nYes or No
                \nEnter answer here:   ",
                  "\nWhat is the average cost per person at the family reunion?
                      \n1 = $50 - $100\n2 $75 - $175\n3 = $25 - $100\n Enter your finale answer here:",
                  "\nNot going to a family reunion can inflict major health and mental issues.\nYes or No:" ]


print("??????????????????????????????????")
print("           Feast your eyes on this                           ")
print("                       nearly immposible quiz                ")
print("??????????????????????????????????")

print("\nThis is a quiz were you will be trying to answer 5 questions about the topic 'Family
    Reunions'")
print("To pass the quiz you must at least asnwer 3 or more questions correctly.")


print("Are you ready?, then let us begin!")



#Copy and pasted quiz questions but each are changed and tweaked. And if the they get the
      question right then the value of the players_score
    is increased by 1

play = True
while play == True:

    players_score = 0

    # Question 1 code
    answer = input(questions[0])
    if answer.upper() == "TRUE" or answer.upper() == "T":
```

```python
        print("\nNice job,that was correct.")
        players_score += 1
    else:
        print("\nNope,the right answer was True.")

    # Question 2 code
    answer = input(questions[1])
    if answer == "5" or answer.upper() == "FIVE":
        print("\ngood one,that was correct.")
        players_score += 1
    else:
        print("\nNot a good one,the correct answer was 5,2,500 people attended.")

    # Question 3 code
    answer = input(questions[2])
    if answer.upper() == "NO" or answer.upper() == "N" or answer.upper() == "FALSE":
        print("\nNot Bad, correct.")
        players_score += 1
    else:
        print("\nFail, the answer was No.")

    # Question 4 code
    answer = input(questions[3])
    if answer == "1" or answer.upper() == 'ONE':
        print("\nYour pretty good,your correct.")
        players_score += 1
    else:
        print("\nWrong,the answer was 1.$50 - $100.")


    # Question 5 code
    answer = input(questions[4])
    if answer.upper() == "YES" or answer.upper() == "Y" or answer.upper() == "TRUE":
        print("\nNice work, you got the question correct.")
        players_score += 1
    else:
        print("\nWrong,the answer was Yes.")


#This tallies up the score, displaying it too the player and telling if the player has passed the quiz or not

    print("\nYour final score is " + str(players_score))
    if players_score < PASS:
        print("Bad news but you failed the quiz, you need at least 3 or more correct answers to pass")
    elif players_score == FLAWLESS:
        print("That was amazing how you answered each question correct.")
    else:
        print("Smart, you passed the quiz")

# Gives the player a chance if they want to play the quiz again until they don't want to no more
    play_again = input("\nDo you you wish to play again,Y/N: ")
    if play_again.upper() == "Y" or play_again.upper() == "YES":
        continue
    else:
        play =False 96

print("\nThank you for participating in my quiz, have a fine rest of your day...")
```

Test table V1.0

Question 1 input

Test data	Expected result	Actual result	Notes
T	Incorrect message	As expected	(Case 1)
true	Correct message	As expected	
f	Incorrect message	As expected	
false	Incorrect message	As expected	
123	Incorrect message	As expected	
hello	Incorrect message	As expected	
Blank	Incorrect message	As expected	(Case 2)

Question 2 input

Test data	Expected result	Actual result	Notes
5	Correct message	As expected	
3	Incorrect message	As expected	
123	Incorrect message	As expected	
7	Incorrect message	As expected	Could fix this in v1.1 checking what the boundaries are of the multi answer questions
2+3	Incorrect message	As expected	
Blank	Incorrect message	As expected	(Case 2)

Question 3 input

Test data	Expected result	Actual result	Notes
n	Incorrect message	As expected	
no	Correct message	As expected	
false	Incorrect message	As expected	Even though the answer was technically correct,I could fix this in v1.1. (Case 1)
123	Incorrect message	As expected	
Yes	Incorrect message	As expected	

Blank	Incorrect message	As expected	(Case 2)

Question 4 input

Test data	Expected result	Actual Results	Notes
3	Incorrect message	As expected	
1	Correct message	As expected	
6	Incorrect message	As expected	
one	Incorrect message	As expected	
123	Incorrect message	As expected	
Blank	Incorrect message	As expected	(Case 2)

Question 5 Input

Test data	Expected result	Actual Results	Notes
yes	Correct message	As expected	
YES	Correct message	As expected	
y	Incorrect message	As expected	

8	Incorrect message	As expected	Same case with question 3 input test data “false” notes (Case 1)
Blank	Incorrect message	As expected	Maybe give the user a chance to answer questions again after entering an input not even close to the answers displayed? (Case 2)

MyQuiz v1.1(Same test data as v1.0)
Question 1 input

Test data	Expected result	Actual result	Notes
T	Correct message	As expected	
true	Correct message	As expected	
f	Incorrect message	As expected	
yes	Incorrect message	As expected	
123	Incorrect message	As expected	
hello	Incorrect message	As expected	
Blank	Incorrect message	As expected	

Question 2 input

Test data	Expected result	Actual result	Notes

5	Correct message	As expected	
3	Incorrect message	As expected	
123	Incorrect message	As expected	
7	Incorrect message	As expected	
2+3	Incorrect message	As expected	
Blank	Incorrect message	As expected	

Question 3 input

Test data	**Expected result**	**Actual result**	**Notes**
n	Correct message	As expected	
no	Correct message	As expected	
false	Correct message	As expected	
123	Incorrect message	As expected	
Yes	Incorrect message	As expected	
Blank	Incorrect message	As expected	

Question 4 input

Test data	Expected result	Actual Results	Notes
3	Incorrect message	As expected	
1	Correct message	As expected	
6	Incorrect message	As expected	
one	Correct message	As expected	
123	Incorrect message	As expected	
Blank	Incorrect message	As expected	

Question 5 Input

Test data	Expected result	Actual Results	Notes
yes	Correct message	As expected	
YES	Correct message	As expected	
y	Correct message	As expected	
8	Incorrect message	As expected	
Blank	Incorrect message	As expected	

Players score messages

Test data	Expected result	Actual Results	Notes
0	Fail message	As expected	
1	Fail message	As expected	
2	Fail message	As expected	
3	Pass message	As expected	
4	Pass message	As expected	
5	Flawless message	As expected	

Grade: Excellence

For Excellence, the student needs to create a flexible and robust computer program.

This involves using conditions and control structures effectively and using constants, variables, or derived values in place of literals to make the program flexible. Testing and debugging the program is required to ensure it works on expected, boundary, and invalid cases.

This student has used ‘methods’ to ensure the code is structured effectively and minimise repeated code. A ‘for’ loop is used to cycle through the questions, allowing additional questions to be easily added and meeting the requirement for flexibility.

Testing and debugging for invalid values can be seen in the testing table. For example, filling the entry fields with long strings of ‘b’ and changing the program when an error was found. ‘If’ statements are used to validate input to ensure invalid cases are handled correctly.

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.

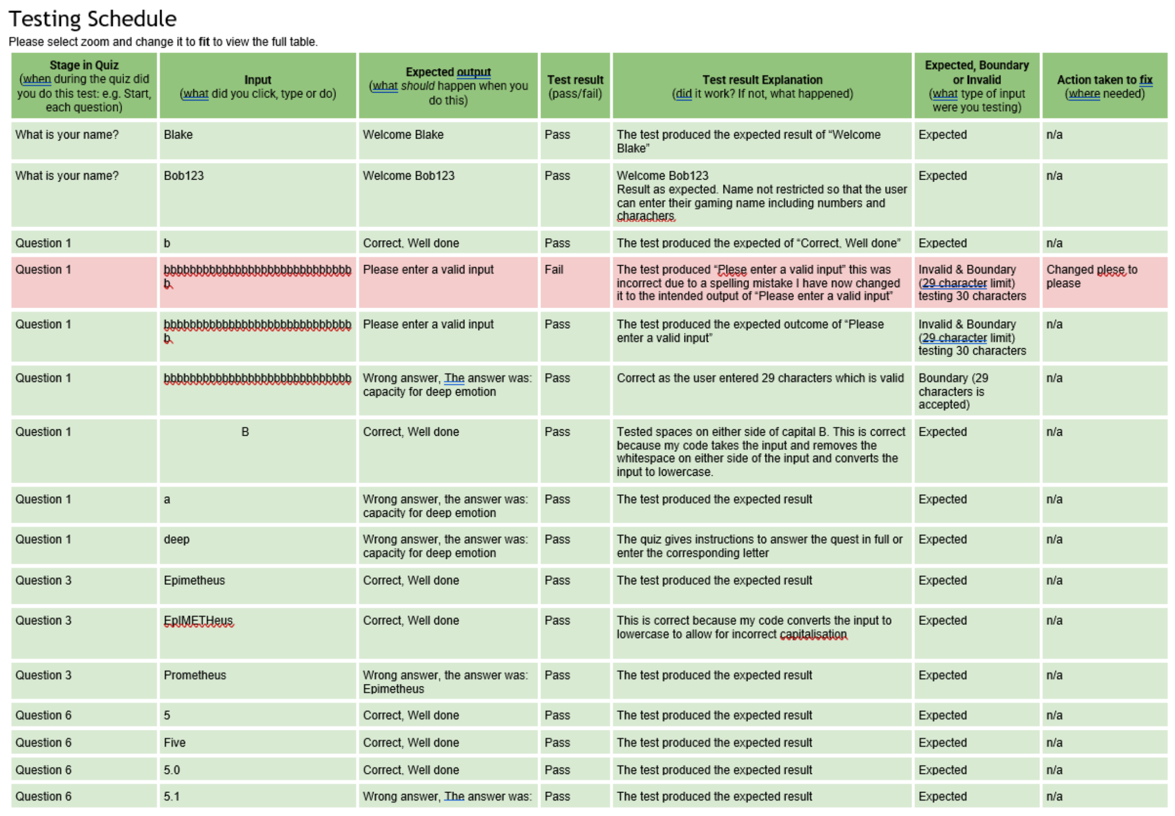
Style of question (e.g. multiple choice, short answer)
A mixture of multiple choice and short answer questions

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.

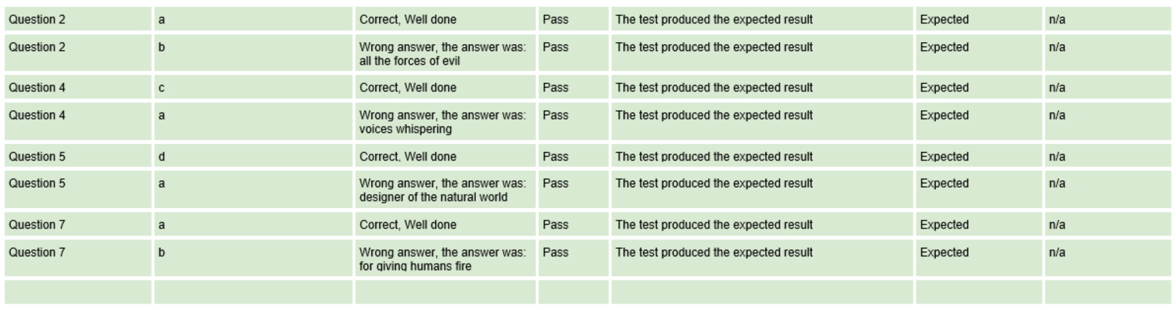
Boundary conditions I could test
If a user doesn't enter an answer, prompt them for an input. If the user enters more than 30 characters prompt them for a correct answer.



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

Stage in Quiz (when during the quiz did you do this test: e.g. Start, each question)	**Input** (what did you click, type or do)	**Expected output** (what *should* happen when you do this)	**Test result** (pass/fail)	**Test result Explanation** (did it work? If not, what happened)	**Expected, Boundary or Invalid** (what type of input were you testing)	**Action taken to fix** (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 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 Ask the question again	Fail	Moved on to the next question. 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 Ask the question again	Pass	The test produced the expected result	Invalid	n/a
End Do you want to play again?	i	Invalid input	Fail	Game ends	Invalid	Add a condition to check for a valid input
End Do you want to play again?	no	End game	Fail	Invalid input please enter yes or no 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 Do you want to play again?	i	Invalid input please enter yes or no Ask the question again	Pass	Print: Invalid input please enter yes or no Ask question again	Invalid	n/a
End Do you want to play again?	Yes	Game restarts	Pass	The test produced the expected result	Expected	n/a
End 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

```python
#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"]),
        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)
```