PEMDAS Validation Testing

# Requirements

Develop an appropriate set of test data to fully validate the program against.

* + the data sets you tested against
  + a brief written explanation of the results of your tests and what you had to fix
  + screen shots of your good test data working

# Test Categories and Cases

1. Test category – score
   1. Test case – score updated on first attempt correct within 1 minute
      1. Expected – 50 points added to score
      2. Passed
   2. Test case – score updated on second attempt correct within 1 minute
      1. Expected – 30 points added to score
      2. Passed
   3. Test case – score updated on third attempt correct within 1 minute
      1. Expected – 10 points added to score
      2. Passed
   4. Test case – score updated on first attempt correct within 2 minutes
      1. Expected – 25 points added to score
      2. Passed – screenshot #1
2. Test category – timer
   1. Test case – timer starts when GET button pressed
      1. Expected – timer starts
      2. Passed
   2. Test case - timer resets to 0 when SUBMIT button pressed and answer is correct
      1. Expected – timer = 0
      2. Passed
   3. Test case – timer keeps running when SUBMIT button pressed and answer is incorrect
      1. Expected – time keeps running
      2. Passed – screenshot #2
   4. Test case – timer resets to 0 when RESET button pressed
      1. Expected – time = 0
      2. Passed
3. Test category – lives
   1. Test case – lives decremented by 1 after three incorrect answers
      1. Expected – lives -= 1
      2. Passed – screenshot #3
   2. Test case – lives reset to 3 when RESET button pressed
      1. Expected – lives = 3
      2. Passed
4. Test category – reset button
   1. Test case – RESET button works if pressed before any other user interaction
      1. Expected – all fields reset
      2. Failed – received exception for trying to cancel timer that was not yet declared
      3. Fixed – added try/except to timer cancel logic – screenshot #4
   2. Test case – RESET button works after a correct answer
      1. Expected – all fields reset
      2. Passed
   3. Test case – RESET button works after an incorrect answer
      1. Expected – all fields reset
      2. Passed
   4. Test case – RESET button works when pressed consecutively
      1. Expected – all fields reset
      2. Passed
5. Test category – exit button
   1. Test case – EXIT button works if pressed before any other user interaction
      1. Passed
   2. Test case – EXIT button works after a correct answer
      1. Expected – application shuts down
      2. Passed
   3. Test case – EXIT button works after an incorrect answer
      1. Expected – application shuts down
      2. Passed
6. Test category – high score
   1. Test case – high score updated after user wins
      1. Expected – score placed in high score
      2. Passed
   2. Test case – high score updated after user loses
      1. Expected – score placed in high score
      2. Passed – screenshot #5
   3. Test case – high score not updated when RESET button pressed
      1. Expected – high score not updated
      2. Passed
7. Test category – get/submit expression
   1. Test case – once GET button pressed, it is disabled and SUBMIT button is enabled
      1. Expected – GET button enabled, SUBMIT button enabled
      2. Passed
   2. Test case – expression generated is syntactically correct
      1. Expected – generated expression can be evaluated
      2. Passed
   3. Test case – number of operators incremented after correct answer
      1. Expected – number of operators incremented
      2. Passed
   4. Test case – number of operators decremented after 3 incorrect answers
      1. Expected – number of operators incremented
      2. Passed
   5. Test case – After a correct answer, GET button is enabled and SUBMIT button is disabled
      1. Expected - GET button enabled, SUBMIT button enabled
      2. Passed
   6. Test case – After first two incorrect answers, GET button is disabled and SUBMIT button is enabled
      1. Expected - GET button enabled, SUBMIT button enabled
      2. Passed
   7. Test case – After third incorrect answer, GET button is disabled and SUBMIT button is enabled
      1. Expected – Popup window gives correct answer
      2. Passed – screenshot #6
   8. Test case – Encouraging popup window shown after 4 and 7 correct answers
      1. Expected – popup window appears
      2. Passed – screenshot #7
8. Test category – win
   1. Test case – winning popup window displayed after 10 correct answers and game resets when popup window closed
      1. Expected - winning popup window displayed after 10 correct answers and game resets when popup window closed
      2. Failed – winning popup window displayed after 11 correct answers
      3. Fixed
         1. Decided to leave 11 as the threshold for winning
         2. Changed comments in program to reflect that change
9. Test category – lose
   1. Test case – study up losing popup window displayed when lives = 0 and score < 200; game resets when popup window closed
      1. Expected - losing popup window displayed when lives = 0 and game resets when popup window closed
      2. Passed – screenshot #9
   2. Test case – nice effort losing popup window displayed when lives = 0 and score >= 200; game resets when popup window closed
      1. Expected - losing popup window displayed when lives = 0 and game resets when popup window closed
      2. Passed – screenshot #10
10. Test category – score after changes
    1. Test case – changed scoring calculation to include multiplication by the number of operators in expression
       1. Expected – 100 points added to score after correct answer on first attempt with two operators
       2. Passed – screenshot #11
11. Test category – invalid answer type
    1. Test case – enter non-numeric answer
       1. Expected – error popup window
       2. Failed – screenshot #12
       3. Fixed - initialized local variable in except portion of try/except statement to avoid exception in later logic
    2. Test case – enter floating point answer
       1. Expected – error popup window
       2. Passed

# Screenshots

Graphical user interface, application

Description automatically generated

## Screenshot #1

Graphical user interface, application

Description automatically generated

## Screenshot #2

Graphical user interface, application

Description automatically generated

## Screenshot #3

## Graphical user interface, application, PowerPoint Description automatically generated

## Screenshot #4

## Graphical user interface, application Description automatically generated

## Screenshot #5

## Graphical user interface, application Description automatically generated

## Screenshot #6

## Graphical user interface, application Description automatically generated

## Screenshot #7

## Screenshot #8

## Graphical user interface, application Description automatically generated

## Screenshot #9

## Graphical user interface, application, PowerPoint Description automatically generated

## Screenshot #10

## Graphical user interface, application Description automatically generated

## Screenshot #11

## Graphical user interface, application Description automatically generated

## Screenshot #12