

IFCS1–Luke Tomlinson

Summative 1: Task 2

LINEAR EQUATION TEST with a Python Script $+$ $-$ $+$

User Documentation

Overview

The project provides a Python-based tool to test your knowledge and understanding of simple linear equations. This guide will help you set up and run the linear equation test on your local machine.

Prerequisites

Before you begin, make sure the following are installed:

- **Python 3.13**

Installation Steps

1. Install Python 3.x
 - Windows: Download from python.org
 - macOS: Download from python.org

After downloading, run the installer and follow the on-screen instructions. Make sure to tick the box that says "Add Python to PATH" during installation.

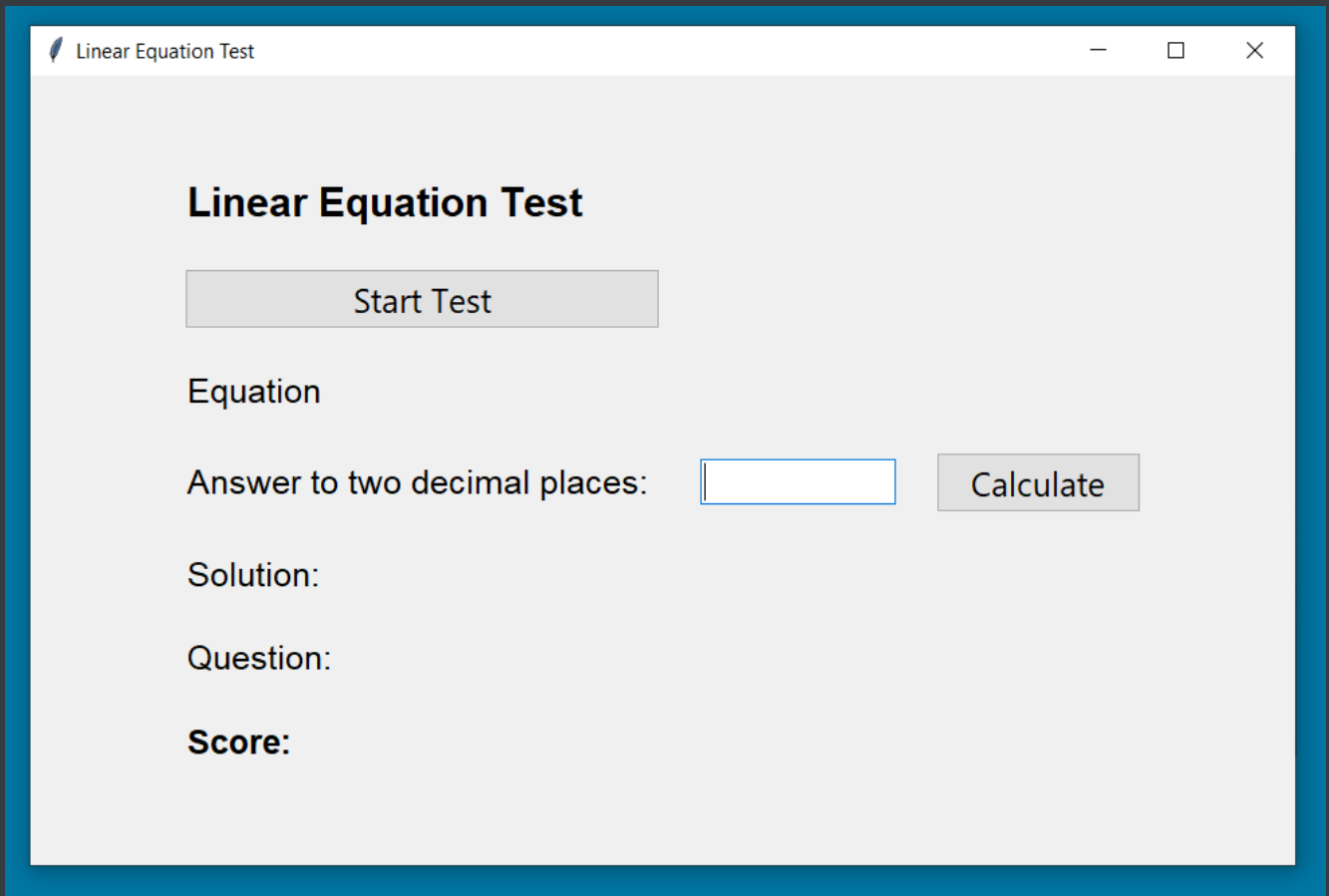
How to Run This Test 🏃🏃🏃

- Open a terminal or command prompt.
- Navigate to the folder where you saved the file.
- Run the script by typing:

```
python3 linear_equation_test.py
```

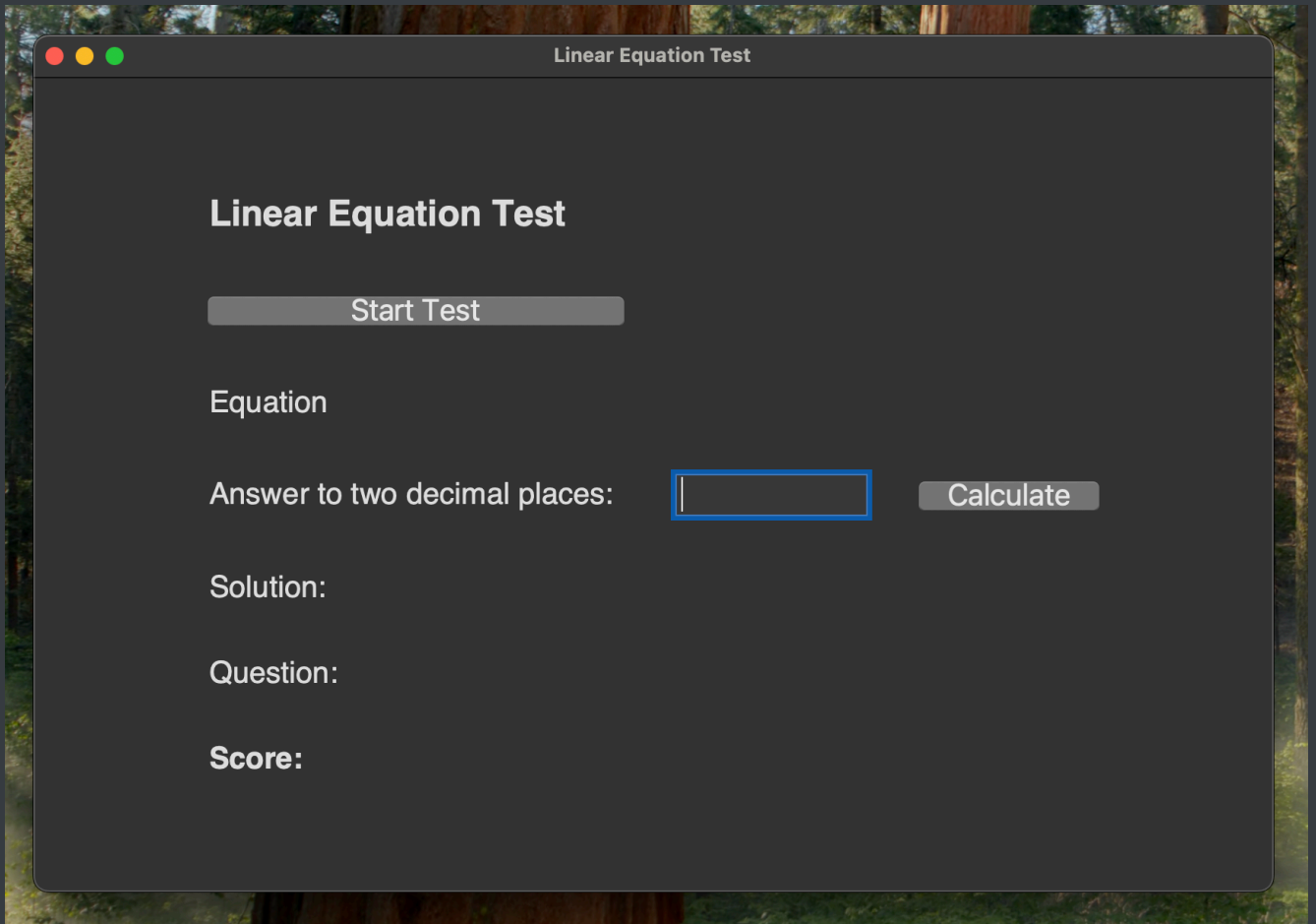
You should have the following tkinter GUI pop up:

- Windows:



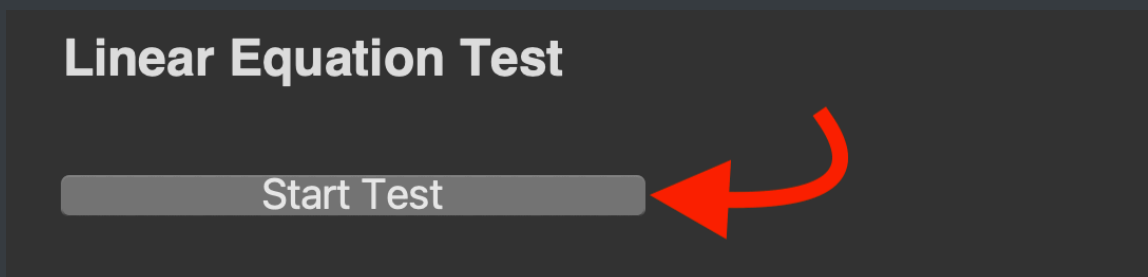
The screenshot shows a tkinter window titled "Linear Equation Test". The window has a white background and a blue border. Inside the window, the title "Linear Equation Test" is displayed in bold black text. Below the title is a grey button labeled "Start Test". Underneath the button is the label "Equation". Below that is the label "Answer to two decimal places:" followed by a text input field and a grey button labeled "Calculate". Below the input field and button are the labels "Solution:", "Question:", and "Score:" in bold black text.

■ macOS:



To save duplicating information moving forward the guide will concentrate on the mac version of the GUI but every action is the same in the Windows version.

To begin click on the `Start Test` button or hit the Return key:



You should see a randomly generated simple equation pop-up:

Linear Equation Test

Start Test

Equation

$$5x = 40$$

Answer to two decimal places:

Calculate

Solution:

Question:

$$1/5$$

Score:

Enter your best answer to two decimal places or less in the answer box:

Equation

$$5x = 40$$

Answer to two decimal places:

8

Calculate

Solution:

Question:

$$1/5$$

Score:

Click **Calculate** button or hit the Return key to check your answer:

Equation	$5x = 40$	
Answer to two decimal places:	<input type="text" value="8"/>	<button>Calculate</button>
Solution:		
Question:	$1/5$	
Score:		


If the answer given is correct the following message is displayed for 4 seconds:

Answer to two decimal places:	<input type="text" value="17"/>	<button>Calculate</button>
Solution:	Correct! 🙌🙌🙌	
Question:	$3/5$	
Score:	66%	

If the answer proves to be incorrect the following message is displayed for 4 seconds:

Answer to two decimal places:	<input type="text" value="5"/>	<button>Calculate</button>
Solution:	The answer is 5.67 Better luck next time! 🍀🍀🍀	
Question:	$2/5$	
Score:	50%	

If no valid answer is entered a message box will appear:



Please enter a valid number

OK

5x + 8 = 16

Calculate

After completing 5 questions a message box will appear with your final score:

Linear Equation Test

Start Test


Equation


3x - 9 = 23

Answer to two decimal places:


10.66

Solution:





Test complete



You scored 2 out of 5 correct.
Final score: 40%

OK

To take another test just repeat the process:

Linear Equation Test

Start Test

Equation

$3x - 9 = 23$

Answer to two decimal places:

Calculate

Solution:

The answer is 10.67
Better luck next time! 🍀🍀🍀

Question:

5/5

Score:

Technical Documentation

Project Structure

```
IFCS1-LT-Sum1/Summative-1-Task-1/  
├─ linear_equation_test_app.py      # Main script for linear  
equation test app
```

linear_equation_test_app.py

This script creates a simple GUI-based test for solving basic linear equations using the tkinter module. It includes:

A DPI-awareness function for improved display scaling on Windows.

A main application class `LinearEquationTest` that initialises and displays the test window.

A Test frame class that manages question generation, input validation, scoring, and user feedback.

Key Features:

DPI Support

A helper function `set_dpi_awareness()` ensures clear visuals on high-resolution screens.

GUI Construction with Tkinter

The application window is built using `tkinter.Tk`, and UI components (labels, buttons, entries) are arranged with grid layout.

Linear Equation Quiz

Questions are randomly generated in one of three formats:

- $ax = b$
- $ax + c = b$
- $ax - c = b$

The user is prompted to solve for x .

Test Control

The user can start the test with a button or by pressing the Enter key. Answers are checked for correctness, and feedback is shown with emojis for encouragement.

Scoring System

Each correct answer increases the score, and the final percentage is displayed when the test is complete.

Error Handling

The `check_answer()` method uses a `try/except` block to catch and handle invalid input (non-numeric answers).

Example Flow:

1 - Launch the app.

2 - Click "Start Test" to begin.

3 - Solve the equation shown.

4 - Enter the answer rounded to two decimal places.

5 - Receive feedback and see progress.

6 - Final results are shown after completing the set number of questions (default: 5).

Dependencies

- The script uses only Python's built-in libraries: `tkinter` and `random` .
- No external packages or installations are required.

Links

tkinter: python.org.