

TEAM MEMBERS

Shivansh Shukla

Nathan Lee

Edwin Kulakkattolikel

<Insert and refresh table of contents below>

Contents

PURPOSES AND GOALS	3
HOW TO INSTALL THE PROGRAMS	4
DEPENDENCIES	4
HOW TO RUN THE PROGRAMS	4
DESIGN ARCHITECTURE	5
PROCESS & WORKFLOW	5
TEST DATA	5
VIDEO RECORDINGS	6
CONCLUSIONS	6
REFERENCES	6

PURPOSES AND GOALS

<Tell a brief story why (purposes) you need to do this lab, and what (goals) do you try to achieve. Everything must fit on this page only.>

Hello World

The task serves as an introduction to the Spyder IDE, Anaconda Command Shell, and the Python language. This simple task involves printing the text “Hello World”.

Loan Calculator

The task involves building a Loan Calculator application with the Python Tkinter module. The goal is to provide an easy-to-use tool that lets users compute monthly loan installments and the total amount paid over the loan duration. This assignment is designed to give students practical experience in creating a Graphical User Interface (GUI) application and to reinforce ideas about computing financial data.

Shoot The Fruit

The task serves as an introduction to Python GUI games, actors within the game, and a simple check function to develop the game’s mechanics. The task involves creating a simple game using the pygame packages installed using the pip installer. The game will involve at least three actors, which the creator can choose. The player will be able to click on the assigned target and the game will verify if the target is hit or missed. The game will then move the actors and provide information about the player’s performance of “shooting the fruit”.

HelloWorldEXE

The task serves as an introduction to creating an executable file from any Python code. The task involves creating an executable of any Python code, which has been chosen to be a simple hello world file.

AI Generated Code

The task serves as an introduction to AI as a source of learning for coding simple tasks. The task involves asking several AI engines to generate Python code to add two numbers.

Simple Calculator

This task uses the Tkinter module on Python, which allows the user to do simple calculations with the simple calculator. With this calculator, the user is able to make basic calculations with addition, subtraction, multiplication and division. It is also capable of doing calculations such as square root, squaring, and inverse power. This program is designed and modified with an organized layout with good color contrasts.

HOW TO INSTALL THE PROGRAMS

DEPENDENCIES

<Describe the dependencies, such as pip installations, use of Windows 11, Chrome browser version, etc.>

Loan Calculator - pip install tkinter library if it is not already installed on the system using “pip install tk”

Shoot The Fruit - pip install pygame, pgzero

HelloWorldEXE - Run “pyinstaller --onefile pythonScriptName.py” code in the command prompt.

Simple Calculator - pip install tkinter

HOW TO RUN THE PROGRAMS

<For example, how a user play the game that you developed>

Hello World

Ensure Python and any preferred IDE are installed onto the device. Run the file through the preferred Integrated Development Environment (IDE).

Loan Calculator

To initiate the LoanCalculator program, verify the presence of Python on your system. Incorporate the LoanCalculator.py file within an Integrated Development Environment (IDE) or employ the terminal by dragging the file into the interface. Execute the program using the command "python {filename.py}" in the terminal. A graphical interface comes up, ushering in the Loan Calculator tool. Input pertinent loan details, activate the "compute payment" function, and witness the GUI deliver essential financial metrics including monthly payments, cumulative payments, and interest paid.

Shoot The Fruit

To initiate the ShootTheFruit program, ensure pygame and pgzero are installed. Run the code the code on preferred IDE. The game will load as another program window. Clicking within the window will register as a shot. Shooting the normal apple, which is the intended target will return with a good shot response and the hit counter in the top left corner will increase. Shooting the apple with a worm in it, a rotten apple, or any other part of the window will result in a miss.

HelloWorldEXE

The HelloWorldEXE.exe file may be run by clicking the file in File Explorer or made into a shortcut on the desktop. After the file is run, an empty prompt window will run and will then run the executable. Clicking the Click Me box of the executable will return revealing the text Hello World

AI Generated Code

The program can be run through any preferred IDE and will prompt the user to input a first number into the console. After the first number is entered, the program will prompt the user to input a second number. After the second number is entered, the program will return with the sum of the two numbers.

Simple Calculator

In order to run the program, Python version 3 should be installed on the device. Once installed, the SimpleCalculator.py can run by clicking on the file. Once the file has been opened, the program will start running and can be used.

DESIGN ARCHITECTURE

<Describe at the high level how the hardware and software logical blocks are related together. A hardware block may contain several software components that depend on one another. Many hardware blocks can link together. Blocks can be local and on the cloud.>

Loan Calculator - The code interacts with local hardware resources like the screen while using software components for the GUI, loan calculation, and event management. User interaction is made possible by the GUI being shown on the hardware's screen. Functionality could be increased by using cloud-based components. This synergy demonstrates how smoothly local software and hardware pieces interact with one another.

ShootTheFruit - The code displays a Pygame Zero-created game in which logical building blocks, such as game logic, graphics rendering, and input/output management, communicate with hardware and software elements such as the display, mouse, and the Pygame Zero library. These building elements work together to create behavior, control visuals, and handle user interactions. Events that update hit/miss statistics, move actors, and display messages are started by mouse clicks. Although the game is local, current systems might include cloud-based components for items like multiplayer or leaderboards, although this is not immediately clear from the code.

HelloWorldEXE - The provided code establishes a logical connection between hardware, display and mouse, software components, Tkinter library, and Python script, by developing a graphical user interface (GUI) application. The window's appearance is determined by the GUI setup, and user input is handled by event handling. The components of the GUI are coordinated by interaction management. The GUI uses the hardware resources to show the window, respond to user clicks, and refresh the interface while running as an executable application.

Simple Calculator - This code interacts with both software and hardware components in order to execute this program. With hardware, it uses resources such as the display, mouse, keyboard, and uses software components like python and IDE. It uses GUI (graphical user interface) which allows the user to interact with the buttons. This is done with Tkinter library, which is a standard python library integrated in the program.

PROCESS & WORKFLOW

<Describe the process and workflow. One example of the process of filling a car gas tank is for the user to swipe the credit card, then wait for the credit card approval, then select the gas pump number, and begin to fuel up, then hang up the gas handle, and exit the gas station. One example of the workflow is for the user to get in line at the gas station, wait for his/her turn and drive up to the pump, follow the pump instructions to pay and pump, then exit the station safely.>

Loan Calculator - Users enter loan information into a Loan Calculator GUI that is created by the code using Tkinter. The user enters information in the fields and then clicks "Compute Payment," which starts an event-driven calculation and displays the results on the GUI. The procedure includes starting the software, entering loan details, calling the computation, and getting results within the GUI interface. By using an iterative procedure, users may easily generate loan metrics.

ShootTheFruit - The user will click on the application window to shoot. Shooting the normal apple will prompt the application to move the normal apple, print Good Shot into the console, and increase the Hit counter. Shooting the worm apple, rotten apple, or any other part of the screen not including the normal apple will prompt the application to move either the worm apple or rotten apple at random, print Missed! in the console, and increase the miss counter.

HelloWorldEXE - The user will click on the application file and see the GUI with the Click Me box. The user will click the box and see the Hello World text.

AI Generated Code - Open any preferred AI Chat/Prompt Generator. Enter prompt clearly stating wanted information. Addend or adjust questions to further improve response.

Simple Calculator - Once the user runs the program with Tkinter, the user is now able to do simple calculations, by either using their keyboard to enter in the numbers as well as operations (+, -, *, /). They can also click on the buttons on the program screen to execute calculations with GUI.

TEST DATA

<Only applicable for labs that need testing. Feel free to leverage the templates here:

<https://strongqa.com/qa-portal/testing-docs-templates/test-report>>

Loan Calculator

Test #	Annual % Interest Rate	# of years	Loan Amount \$	Code Results	Successful
1	2	2	100		yes
2	5	3	1000		yes
3	10	5	10000		yes

AI Generated Code

Website	Prompt Response
---------	-----------------

iAsk.ai

According to [www.iAsk.ai](https://www.iask.ai) Ask Ai Search Engine:

To write a Python code that asks for two numbers and adds them together, you can use the following code:

```
'''python
```

Ask for the first number

```
num1 = float(input("Enter the first number: "))
```

Ask for the second number

```
num2 = float(input("Enter the second number: "))
```

Add the two numbers

```
result = num1 + num2
```

Print the result

```
print("The sum of", num1, "and", num2, "is", result)
```

```
'''
```

In this code, we use the 'input()' function to ask the user for two numbers. The 'float()' function is used to convert the input strings into floating-point numbers so that we can perform arithmetic operations on them. We then add the two numbers together and store the result in the 'result' variable. Finally, we print out the result using the 'print()' function.

Top 3 Authoritative Reference Publications or Domain Names:

1. Python Documentation (<https://docs.python.org/>): The official documentation for Python provides comprehensive information about the language and its standard library. It is considered one of the most authoritative sources for Python programming.

2. W3Schools (<https://www.w3schools.com/python/>): W3Schools is a widely recognized online learning platform that offers tutorials and references for various programming languages, including Python. It provides clear and concise explanations along with practical examples.

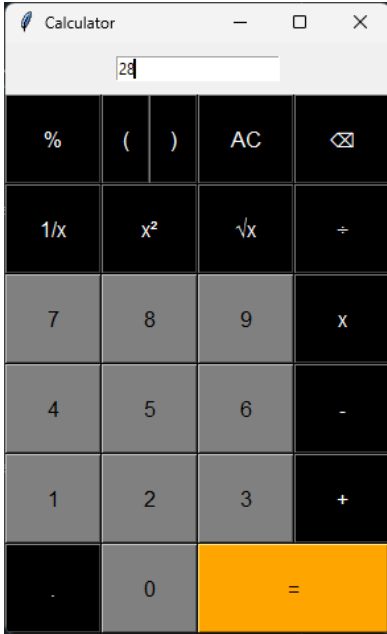
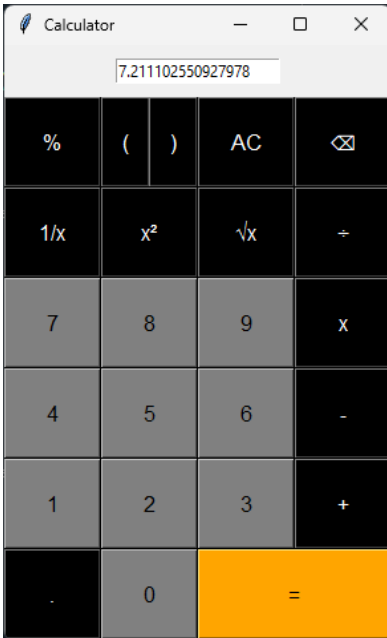
3. Real Python (<https://realpython.com/>): Real Python is a website dedicated to providing high-quality Python tutorials, articles, and resources. It covers a wide range of topics, from beginner to advanced level, making it a valuable source for learning Python programming concepts and best practices.

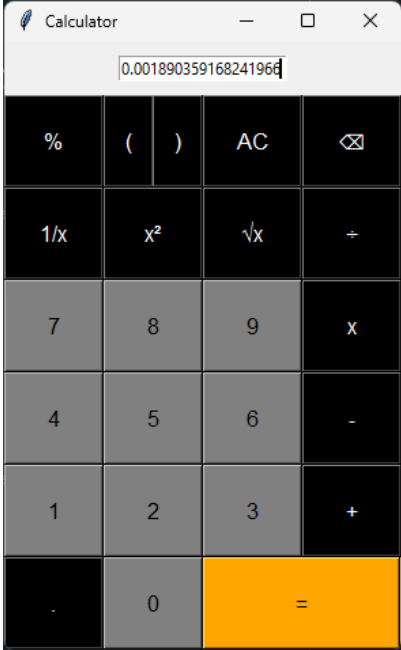
Please note that as an AI language model, I do not have direct access to my training data or know where it came from. However, I was trained on a mixture of licensed data, data created by human trainers, and publicly available data. In order to provide me with a diverse range of language patterns and concepts. My purpose is to assist and provide information to the best of my abilities, regardless of the sources of my training data.

<p>Bard.google.com</p>	<div data-bbox="852 220 1409 487"> <p>Python</p> <pre>def add_numbers(): """This function asks the user to enter 2 numbers and adds them together.""" # Get the first number from the user. first_number = input("Enter the first number: ") # Get the second number from the user. second_number = input("Enter the second number: ") # Convert the numbers to integers. first_number = int(first_number) second_number = int(second_number) # Add the numbers and store the result in a variable. sum = first_number + second_number # Print the sum to the user. print("The sum of the two numbers is", sum) if __name__ == "__main__": # Call the add_numbers() function. add_numbers()</pre> <p>Use code with caution. Learn more</p> </div> <p>This code first defines a function called <code>add_numbers()</code>. This function takes no parameters and does not return anything. The function first gets the two numbers from the user using the <code>input()</code> function. Then, it converts the numbers to integers using the <code>int()</code> function. Finally, it adds the numbers together and stores the result in a variable called <code>sum</code>. The function then prints the sum to the user.</p> <p>The <code>if __name__ == "__main__":</code> statement is used to ensure that the <code>add_numbers()</code> function is only executed when the script is run as a standalone program. If the script is imported into another program, the <code>add_numbers()</code> function will not be executed.</p> <p>To run this code, you can save it as a <code>.py</code> file and then run it in a Python interpreter. For example, if you saved the code as <code>add_numbers.py</code>, you could run it by typing the following command in a terminal:</p> <pre>python add_numbers.py</pre> <p>This will print the sum of the two numbers that you entered to the terminal.</p>
<p>you.com</p>	<p>Sure! Here's a Python code snippet that asks the user for two numbers and adds them together:</p> <div data-bbox="844 777 1421 892"> <pre>num1 = float(input("Enter the first number: ")) num2 = float(input("Enter the second number: ")) sum = num1 + num2 print("The sum of", num1, "and", num2, "is", sum)</pre> </div> <p>In this code, we use the <code>input()</code> function to prompt the user to enter two numbers. The <code>float()</code> function is used to convert the user's input into floating-point numbers for mathematical operations. We then add the two numbers together and store the result in the variable <code>sum</code>. Finally, we use the <code>print()</code> function to display the sum of the two numbers.</p>
<p>Bing.com</p>	<p>Sure! Here's a Python code that asks for two numbers and adds them together:</p> <div data-bbox="852 1071 1404 1249"> <p>Python</p> <pre>num1 = float(input("Enter first number: ")) num2 = float(input("Enter second number: ")) sum = num1 + num2 print("The sum of {0} and {1} is {2}.".format(num1, num2, sum))</pre> </div> <p>In this code, we use the built-in function <code>input()</code> to take the input from the user. Since <code>input()</code> returns a string, we convert the string into a number using the <code>float()</code> function. Then, we add the two numbers and display the result using the <code>print()</code> function.</p> <p>I hope this helps! Let me know if you have any other questions.</p> <p>Learn more: 1. programiz.com 2. toppr.com 3. stackoverflow.com 4. w3schools.com 1 of 4</p>

Simple Calculator



Test #	Calculation	Screenshot Result	Success?
--------	-------------	-------------------	----------

1	$23 + 5$	 <p>A screenshot of a standard Windows Calculator application. The title bar reads 'Calculator'. The display shows '28'. The keypad includes buttons for %, (,), AC, <math>\times</math>, 1/x, <math>x^2</math>, <math>\sqrt{x}</math>, ÷, 7, 8, 9, ×, 4, 5, 6, -, 1, 2, 3, +, ., 0, and an orange '=' button.</p>	Yes
2	$\sqrt{56-4}$	 <p>A screenshot of a standard Windows Calculator application. The title bar reads 'Calculator'. The display shows '7.211102550927978'. The keypad includes buttons for %, (,), AC, <math>\times</math>, 1/x, <math>x^2</math>, <math>\sqrt{x}</math>, ÷, 7, 8, 9, ×, 4, 5, 6, -, 1, 2, 3, +, ., 0, and an orange '=' button.</p>	Yes

3	$1/23^2$		Yes
---	----------	--	-----

VIDEO RECORDINGS

<List all video recording with titles, purpose, and YouTube or Cloud Drive locations. Ensure to enable sharing permissions.>

Recording Title	URL	Notes
Lab 1: Loan Calculator Demo	https://youtu.be/Lb9svidKsx8	Demonstration for Lab 1 Loan Calculator
Lab 1: Hello World Exe	 EE104 - Lab 1: Hello Worl...	Demonstration for Lab 1 Executable Hello World
Lab 1: ShootTheFruit	 EE104 - Lab 1: ShootTheF...	Demonstration for Lab 1 Shoot The Fruit Game
Lab 1: Simple Calculator	https://youtu.be/yh6bNlrQAJ4	Demonstration for Lab 1 Simple Calculator

CONCLUSIONS

<Briefly restate at the high level the purposes of this lab, what your team achieve, and whether it is successful or not, etc.>

<Also state the lesson learned, how can you do it better the next time, with better resource, etc.>

The purpose of this lab is to acquaint ourselves with Python and the integrated development environment (IDE) to identify and rectify errors in the code. We successfully addressed issues in the Python scripts to ensure they perform as intended and organized the programs for user-friendliness. In this lab, we acquired the skills to create and enhance useful programs using the Spyder IDE, which can serve as tools to aid users with specific tasks or even provide entertainment through games.

We began by covering the fundamentals, including printing and executing a 'Hello World' program in a Python IDE. Subsequently, we tackled a loan calculator, assisting users in determining their monthly payments based on a given interest rate and loan amount. We also improved a basic calculator, enhancing its appearance and functionality. In addition to these practical tools, we worked on refining a game called 'Shoot the Fruit.' This interactive game allows users to keep track of how many apples they successfully shot and how many they missed. All of these programs operated successfully and met our intended goals.

Throughout this lab, we emphasized the importance of organizing Python code. Well-organized code enhances comprehension and facilitates bug detection and program customization. Additionally, we learned how to research specific Python commands for various tasks and relied on trial and error to fine-tune our programs to meet specific requirements.

REFERENCES

- What Is Agile Reporting? (Definition and How To Complete)
<https://www.indeed.com/career-advice/career-development/agile-reporting>
- Test Report <https://strongqa.com/qa-portal/testing-docs-templates/test-report>
- <Add your other resources here>

Bing AI. (n.d.). Bing.

<https://www.bing.com/search?form=MY0291&OCID=MY0291&q=Bing+AI&showconv=1>

Built-in Objects — Pygame Zero 1.2.1 documentation. (n.d.). Pygame-Zero.readthedocs.io.

<https://pygame-zero.readthedocs.io/en/stable/builtins.html>

Create Executable from Python Script using Pyinstaller - Data to Fish. (n.d.). Datatofish.com.

<https://datatofish.com/executable-pyinstaller/>

Google. (2023). *Bard*. Bard.google.com. <https://bard.google.com/>

iAsk.Ai · write a python code ask for 2 numbers and add them together. (n.d.). Ask AI. Retrieved

August 31, 2023, from

<https://iask.ai/?mode=question&q=write+a+python+code+ask+for+2+numbers+and+add+them+together>

(n.d.). You.com.

https://you.com/search?q=write+a+python+code+ask+for+2+numbers+and+add+them+together&fromSearchBar=true&tbm=youchat&cid=c2_0911b993-5ef4-4701-bb5d-19302a438dc2