Lab 11: GUIs

Goals for this lab:

- Construct a Python program that has a Graphical User Interface (GUI)
- Construct a Python program that uses event-driven programming to coordinate and handle user inputs
- Use container widgets, such as `Frames`, to group and display widgets in a GUI
- Use functional widgets, such as `Button`n and `Entry`, to allow the user to interact with a program
- Write and execute Python programs using the given instructions

General Guidelines:

- Use meaningful variable names
- Use appropriate indentation
- Use comments, especially for the header, variables, and blocks of code.

Example Header:

```
# Author: Your name
# Date: Today's date
# Description: A small description in your own words that describes what
the program does. Any additional information required for the program to
execute should also be provided.
```

Example Function Docstring:

```
"""
General function description
:param p1: DESCRIPTION
:type p1: TYPE
:param p2: DESCRIPTION
:type p2: TYPE
:return: DESCRIPTION.
"""
```

1. Basic GUI

Graphical user interfaces or GUIs provide software developers with an expanded set of tools to allow users to interact with their programs. Rather than simply waiting for the user to enter in text to the terminal or console, the user can now click buttons, select items from menus, select one of many options from a selection of check boxes, and enter text into multiple areas before it needs to be processed. The goal now is to obtain a better understanding of how to design and construct a GUI using the widgets provided by Python's tkinter module, and how to handle user inputs via those widgets.

For this exercise you will be writing a program to allow the user to convert distances from meters to smoots and from smoots to meters. A smoot is a unit of length created by the Lambda Chi Alpha fraternity at MIT in 1958 and is roughly 1.7 meters in length. It is based on their at the time pledge, Oliver Smoot's, actual height and was used to measure the length of the Harvard Bridge. Ironically, Oliver went on to become chairman of the American National Standards Institute and later president of the International Organization for Standardization. In any case, your program will provide the user with a brief bit of information about the smoot, one field to enter in a length in smoots, another to enter in a length in meters, a button to convert smoots to meters, another button to convert meters to smoots, and a final button to quit the program.

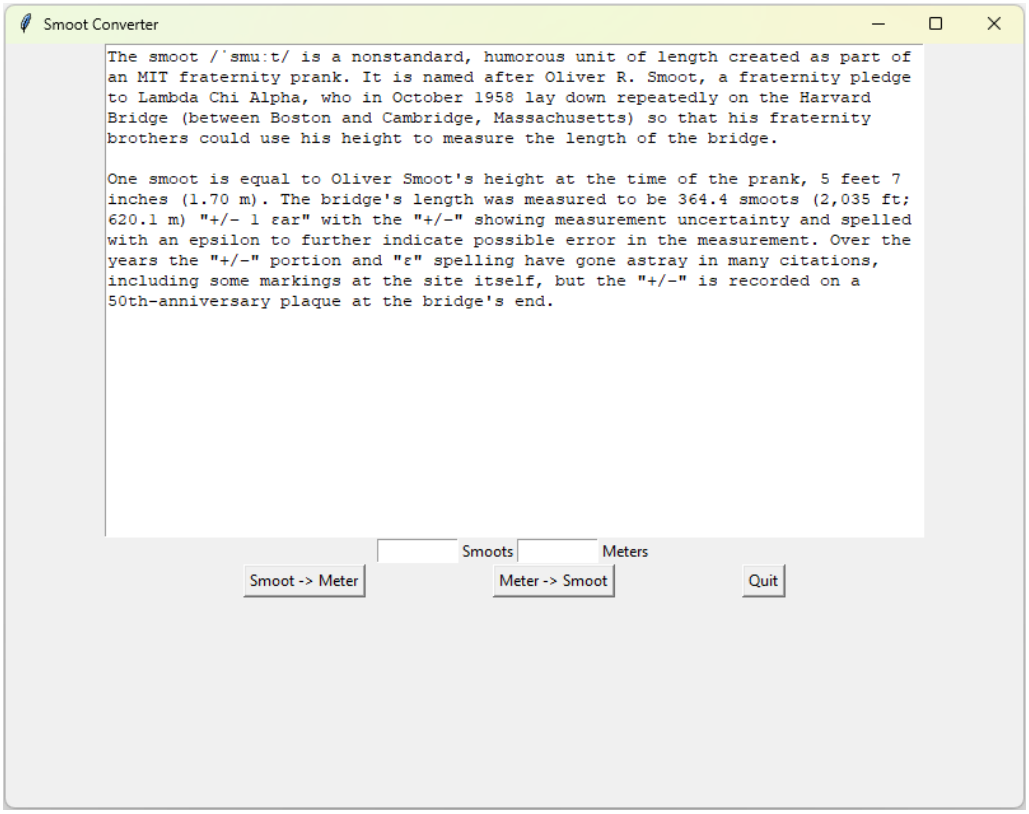
To complete this program:

- In a file named **SmootConverter.py**
    - Define a class named `SmootGUI` which contains:
        - A constructor function that takes in `self`, returns nothing, and should:
            - Create a `Toplevel` widget and store it in a variable named `main_window` to act as the main window for the program
                - Set the title to "Smoot Converter"
                - Set the dimensions to 800 pixels wide by 600 pixels tall
            - Create a variable named `smootInfo` to store the text regarding the history of the smoot
                - The smoot /ˈsmuːt/ is a nonstandard, humorous unit of length created as part of an MIT fraternity prank. It is named after Oliver R. Smoot, a fraternity pledge to Lambda Chi Alpha, who in October 1958 lay down repeatedly on the Harvard Bridge (between Boston and Cambridge, Massachusetts) so that his fraternity brothers could use his height to measure the length of the bridge.

                  One smoot is equal to Oliver Smoot's height at the time of the prank, 5 feet 7 inches (1.70 m). The bridge's length was measured to be 364.4 smoots (2,035 ft; 620.1 m) "+/− 1 ear" with the "+/−" showing measurement uncertainty and spelled with an epsilon to further indicate possible error in the measurement. Over the years the "+/−" portion and "ε" spelling have gone astray in many citations, including some markings at the site itself, but the "+/−" is recorded on a 50th-anniversary plaque at the bridge's end.

- Create a `Text` widget, stored in a variable named `smootTextArea`, whose parent is `main_window` and which uses `WORD` wrapping
    - Using the appropriate variable, insert the text regarding the history of the smoot into the `Text` widget
    - Configure the state of the `Text` widget to be `DISABLED`
- Create a `Frame` widget, stored in a variable named `inputFrame`, to store the user's input, whose parent is the `main_window`
- Create a `Frame` widget, stored in a variable named `buttonFrame`, to store the buttons, whose parent is the `main_window`
- Create an `Entry` widget, stored in a variable named `smootField`, with a width of 10 and a Label widget, stored in a variable named `smootLabel`, with the text Smoots and set their parents to `inputFrame`
- Create an `Entry` widget, stored in a variable named `meterField`, with a width of 10 and a Label widget, stored in a variable named `meterLabel`, with the text Meters and set their parents to `inputFrame`
- Create a Button widget, stored in a variable named `smootMeterButton`, whose parents is `buttonFrame`, text is Smoot -> Meter, and command is `getSmoots`
- Create a Button widget, stored in a variable named `meterSmootButton`, whose parents is `buttonFrame`, text is Meter -> Smoot, and command is `getMeters`
- Create a Button widget, stored in a variable named `quitButton`, whose parents is `buttonFrame`, text is Quit, and command destroys main_window
- Pack `smootField`, `smootLabel`, `meterField`, and `meterLabel`, in that order, setting the side to left
- Pack `smootMeterButton`, `meterSmootButton`, and `quitButton`, in that order, setting the side to left and giving each an external padding of 50 pixels
- Pack `smootTextArea`, `inputFrame`, and `buttonFrame`, in that order, setting the side to top
- Call the `mainloop()` function

- Your final GUI should look similar to the following:



- A `getSmoots()` function that takes in `self`, returns nothing, and should:
  - Get the text from `smootField` and store it in a variable
  - Attempt to convert the value to a floating-point number
    - If it generates an exception, catch the exception and spawn a `showerror messagebox` informing the user that their value is not a valid distance, and immediately exit the function but not the program
    - Otherwise, multiply the value by 1.7018 to convert it to meters
  - Delete any data in `meterField`
  - Insert the calculated value in meters to `meterField`, with two decimals of precision
- A `getMeters()` function that takes in `self`, returns nothing, and should:
  - Get the text from `meterField` and store it in a variable
  - Attempt to convert the value to a floating-point number
    - If it generates an exception, catch the exception and spawn a `showerror messagebox` informing the user that their value is not a valid distance, and immediately exit the function but not the program
    - Otherwise, divide the value by 1.7018 to convert it to smoots
  - Delete any data in `smootField`
  - Insert the calculated value in smoots to `smootField`, with two decimals of precision

- Define a `main()` function that takes in no parameters, returns nothing, and should:
  - Instantiate an instance of your `SmootGUI`
- Call your `main()` function

Complete the program, making sure to execute it to verify that it produces the correct results. Note that you will submit the **SmootConverter.py** file to Canvas.

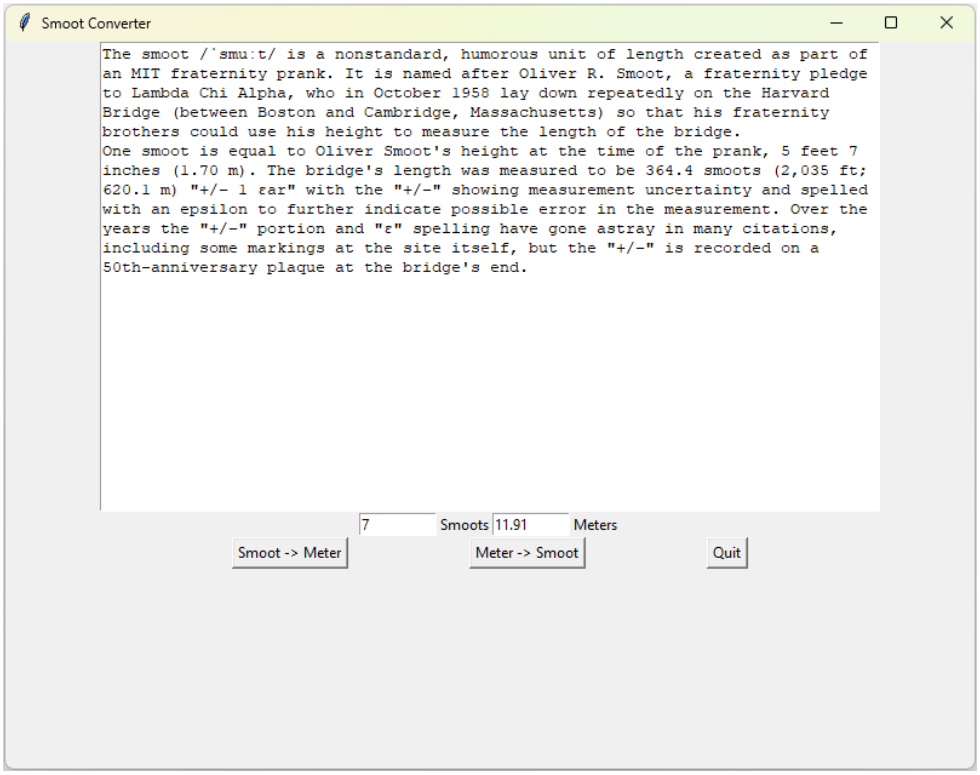
Submission

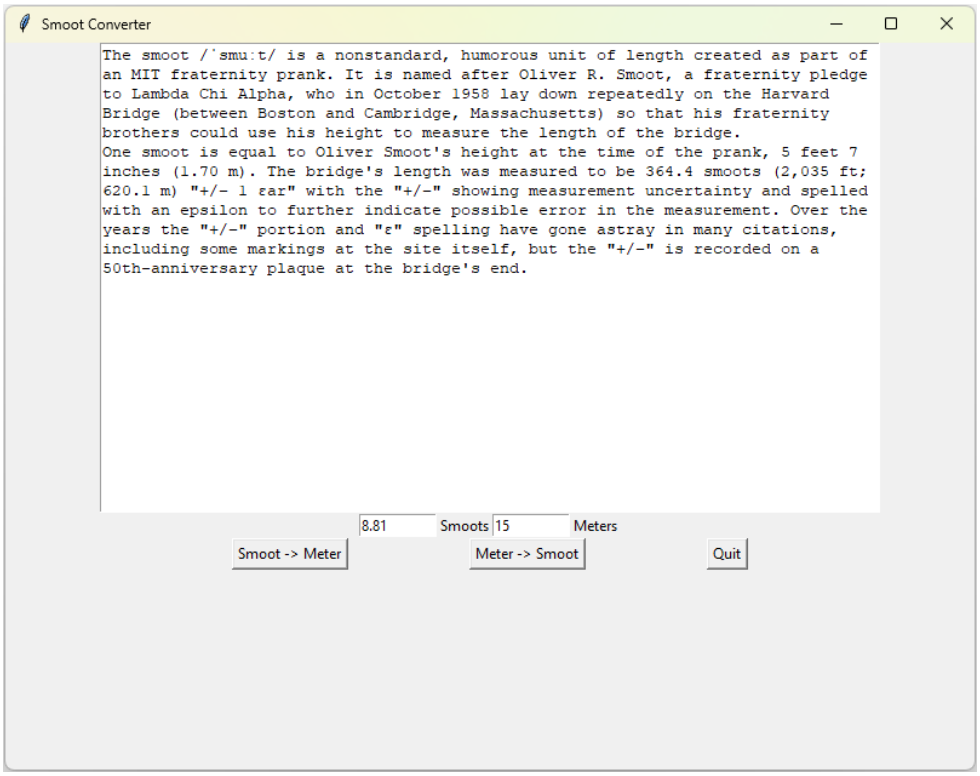
Once you have completed this lab it is time to turn in your work. Please submit the following file to the Lab 11 assignment in Canvas:

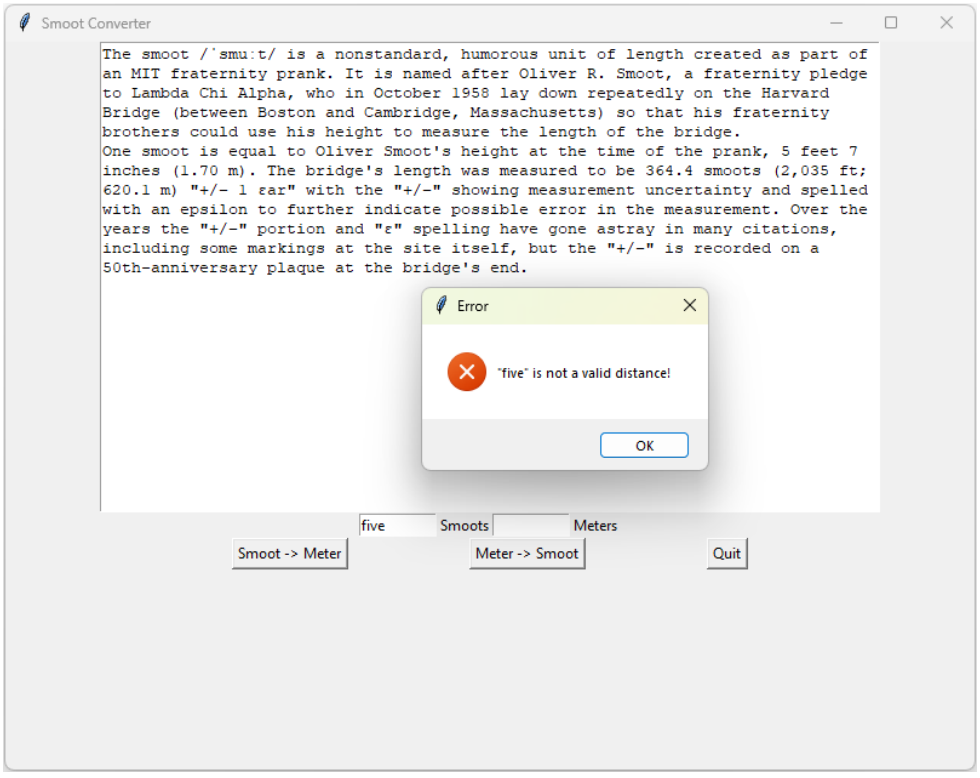
- **SmootConverter.py**

Sample Output

SmootConverter.py




Smoot Converter
The smoot /ˈsmuːt/ is a nonstandard, humorous unit of length created as part of an MIT fraternity prank. It is named after Oliver R. Smoot, a fraternity pledge to Lambda Chi Alpha, who in October 1958 lay down repeatedly on the Harvard Bridge (between Boston and Cambridge, Massachusetts) so that his fraternity brothers could use his height to measure the length of the bridge.
One smoot is equal to Oliver Smoot's height at the time of the prank, 5 feet 7 inches (1.70 m). The bridge's length was measured to be 364.4 smoots (2,035 ft; 620.1 m) "+/− 1 ear" with the "+/−" showing measurement uncertainty and spelled with an epsilon to further indicate possible error in the measurement. Over the years the "+/−" portion and "ε" spelling have gone astray in many citations, including some markings at the site itself, but the "+/−" is recorded on a 50th-anniversary plaque at the bridge's end.
8.81
Smoots
15
Meters
Smoot -> Meter
Meter -> Smoot
Quit


Smoot Converter
The smoot /ˈsmuːt/ is a nonstandard, humorous unit of length created as part of an MIT fraternity prank. It is named after Oliver R. Smoot, a fraternity pledge to Lambda Chi Alpha, who in October 1958 lay down repeatedly on the Harvard Bridge (between Boston and Cambridge, Massachusetts) so that his fraternity brothers could use his height to measure the length of the bridge.
One smoot is equal to Oliver Smoot's height at the time of the prank, 5 feet 7 inches (1.70 m). The bridge's length was measured to be 364.4 smoots (2,035 ft; 620.1 m) "+/− 1 ear" with the "+/−" showing measurement uncertainty and spelled with an epsilon to further indicate possible error in the measurement. Over the years the "+/−" portion and "ε" spelling have gone astray in many citations, including some markings at the site itself, but the "+/−" is recorded on a 50th-anniversary plaque at the bridge's end.
Error
"five" is not a valid distance!
OK
five
Smoots
Meters
Smoot -> Meter
Meter -> Smoot
Quit

Rubric

For each program in this lab, you are expected to make a good faith effort on all requested functionality. Each submitted program will receive a score of either 100, 75, 50, or 0 out of 100 based upon how much of the program you attempted, regardless of whether the final output is correct (See table below). The scores for all of your submitted programs for this lab will be averaged together to calculate your grade for this lab. Keep in mind that labs are practice both for the exams in this course and for coding professionally, so make sure you take the assignments seriously.

Score	Attempted Functionality
100	100% of the functionality was attempted
75	<100% and >=75% of the functionality was attempted
50	<75% and >=50% of the functionality was attempted
0	<50% of the functionality was attempted