12 Team Activity: Using Objects

Instructions

Arrange a one hour synchronous meeting with your team for this activity. Online students should coordinate a video-sharing meeting. Campus students will use class time for this meeting. You should prepare for this meeting by completing the preparation material and the individual checkpoint assignment beforehand.

Problem Statement

Almost all of the programs that you wrote for this course receive input from and print results to a console (also called a terminal) window. However, most users prefer to interact with a program through a graphical user interface (GUI) that contains icons, text fields, drop-down lists, buttons, etc. Within a GUI, the individual components (icons, text fields, etc.) are called widgets. Most libraries for creating GUIs use object oriented programming because each widget is an object with attributes and methods.

Helpful Documentation

GUI Programming with Tkinter

Official documentation for the tkinter module

Assignment

Write a Python program named `gui.py` that gets user input from a GUI, performs a simple calculation, and displays the result in a GUI. Do the following:

1. Download these two Python files: heart_rate.py and number_entry.py and save them in the same folder where you will save your Python program.

2. Open the `heart_rate.py` file in VS Code and run it. Notice how running the program opens a GUI where a user can enter his age and see heart rates for exercising. Experiment with the GUI by entering different ages, seeing the results, and clicking the "Clear" button.

3. Read the comments and examine the code in the `heart_rate.py` program. Notice the following:
   a. To create its GUI, the heart_rate program uses the `tkinter` module which is imported at line 4.
   b. The `main` function begins at line 8.
   c. The `main` function creates a `tk.TK` root object and uses that object to create the main window for the program.
   d. Beginning at line 47, the `populate_main_window` function creates labels, text entry boxes, and buttons and places them in a grid.
   e. Inside the `populate_main_window` function, there are two nested functions named `calc` and `clear`. The `calc` function is called each time the user enters a digit in the age text field. The `clear` function is called each time the user clicks the "Clear" button.

4. Choose a simple calculation or formula for your program to calculate. You could choose one of these:
   a. Area of a circle: $a = \pi r^2$
   b. Swing time of a pendulum: $t = 2\pi \sqrt{\frac{n}{9.81}}$
   c. Area of a rectangle: $a = wh$

d. Volume of a tire: $v = \frac{\pi \, w^2 \, a \left(wa + 2{,}540 \, d \right)}{10{,}000{,}000}$

5. Use the `heart_rate.py` program as a reference (you could even copy and paste parts of it or all of it) and write a Python program with a GUI that calculates the formula that you chose.

Core Requirements

1. Your program must include a GUI that opens when you run your program.
2. The GUI must allow a user to enter input.
3. When the user enters valid input, your program must compute correct results and display those results in the GUI.

Stretch Challenges

If your team finishes the core requirements in less than an hour, complete one or more of these stretch challenges. Note that the stretch challenges are optional.

1. Add a "Clear" button to your GUI that clears all inputs and outputs when the user clicks it.
2. Add a label that acts as a status bar at the bottom of your GUI. Your program should display an error message in the status bar when the user enters invalid input. Your program should clear the status bar when the user enters valid input.

Testing Procedure

Run your program and enter various inputs, including invalid values. Verify that your program doesn't crash, behaves as expected, and displays correct results.

Ponder

After you finish your program, examine the code in the `populate_main_window` function and realize that you wrote object oriented code when you used objects such as *lbl_age*, *lbl_width*, *txt_age*, and *txt_width* and the dot operator to call methods such as `grid`, `config`, `get`, and `clear`.

Sample Solution

Please work diligently with your team for the one hour meeting. After the meeting is over, please compare your approach to the sample solution [↓]. Please ***do not look at the sample solution*** until you have either finished the program or diligently worked for at least one hour. At the end of the hour, if you are still struggling to complete the assignment, you may use the sample solution to help you finish.

Submission

When you have finished the activity, please report your progress via the associated I-Learn quiz. When asked about which of the requirements you completed, feel free to include any work done during the team meeting or after the meeting, including work done with the help of the sample solution, if necessary. In short, report on what you were able to accomplish, regardless of when you completed it or if you needed help from the sample solution.