**Lesson 2 – Every Bit of the Internet**

**ACTIVITY 2.1.4 – Design the User Experience**

**\*\*Instructions:** Please change the text color of your responses to red text. Please organize the endings to each page.

[**4-color window**](https://drive.google.com/file/d/1sowLlDEEHGPR8yz4hhcPs2rGsqBpzNDs/view?usp=sharing)

[**Simple window**](https://drive.google.com/file/d/1IxVDgswqx77v29kSlk0BxZplwxMztXf6/view?usp=sharing)

**VOCABULARY**

Application Programming Interface (API) - A way for programmers to share code with other programmers, such as the methods in a library.

Widgets - A component of a GUI

Arguments - An input value to a function/method that specifies the parameters.

GUI - A way of interacting with a program through visuals

STEP 14: The pack method of widgets has options (keyword arguments) available as well. Add pady=50 to one cell to pack, and padx=50 to the other. What happens?

The label with pady=50 is moved 50 pixels down, and the label with padx=50 is moved 50 pixels to the right.

STEP 19: Add the options (keyword arguments) row and column to frame\_login’s grid call, both with value 0. Also, add the keyword argument sticky with value “news”. This means that when the window is expanded, the frame remains anchored to the North, East, West, and South sides of the window, ensuring that it expands with the window. ***Insert the code that completes this step.***

frame\_login.grid(row=0, column=0, sticky="news")

STEP 22: Create a function just after the import in your program called test\_my\_button with no parameters. In the body of this function, include a call to frame\_auth’s tkraise method with no arguments in the body of your test\_my\_button function. ***Insert the code that completes this step.***

def test\_my\_button():

frame\_auth.tkraise()

STEP 26: Make your GUI display the user's password on frame\_auth. Use the TODOs below as a guide. You may use the *multifactorgui.py* program and any documentation you've found useful on the internet to help you create your code. ***Insert the code that completes this step.***

def check\_login():

entered\_uname = str(ent\_uname.get())

entered\_pwd = str(ent\_pass.get())

if (entered\_uname==uname and entered\_pwd==pwd):

frame\_auth.tkraise()

lbl\_success.config(text=("Password: " + entered\_pwd))

else:

mb.showinfo("Login Failed","Incorrect Credentials Entered")

STEP 28:Use the grid method to recreate the GUI shown below. You may use any documentation you find online to help you with your recreation. ***Insert the code that completes this step.***

import tkinter as tk

root = tk.Tk()

root.wm\_geometry("500x500")

frame\_blue = tk.Frame(root, background="Blue")

frame\_blue.grid(row=0, column=0)

frame\_blue.config(width=350, height=250)

frame\_green = tk.Frame(root, background="Lime")

frame\_green.grid(row=0, column=1)

frame\_green.config(width=150, height=250)

frame\_red = tk.Frame(root, background="Red")

frame\_red.grid(row=1, column=0)

frame\_red.config(width=350, height=250)

frame\_yellow = tk.Frame(root, background="Yellow")

frame\_yellow.grid(row=1, column=1)

frame\_yellow.config(width=150, height=250)

root.mainloop()

*multifactorgui.py* is as much a library as Tkinter is. Compare how each allows for the reuse of code. (Conclusion)

Multifactorgui.py and tkinter both allow for the reuse of code by abstracting it into classes and methods. However, tkinter was originally written in c, while multifactorgui.py is entirely python, although it does have tkinter as a dependency. Additionally, multifactorgui.py is a user created package, while tkinter is one of the default packages included in python.