


## 7OOP FINALS LAB TASK 4 (PAIRED)

### I. PROBLEM



**Marsha Superio**  
Nov 15 (Edited Nov 22)

### FINALS PAIRED Task 1: DEMO Tkinter on Nov 22

**Instructions:**

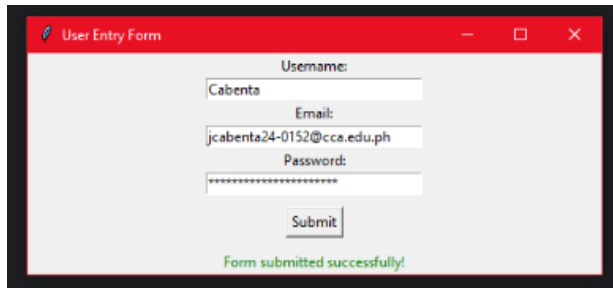
1. Make sure you have a partner to work with (See randomly generated pairs below)
2. You will be assigned a certain Tkinter Elements/Controls and present the ff:  
**Discuss/Present the ff:**
  1. What is the used or **purpose of the widget** or **layout manager** or **container**
  2. What are the methods that you can use to change the look and settings of the widget or control and present the syntax

\*Explore more, do not just settle on the simple examples in the lecture i gave you
3. DEMO the code and mapped the output to the code that you did.

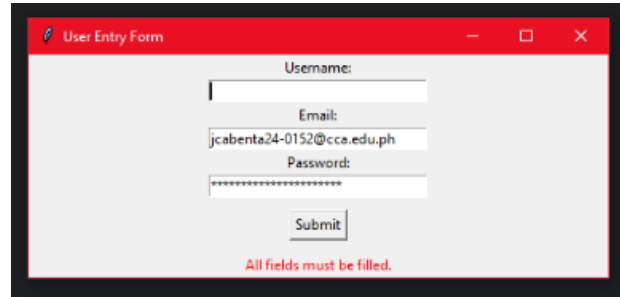
### II. SOURCE CODE

```
try-except.py x
1 from tkinter import *
2 import re # for email pattern checking
3
4 def submit_form():
5     username = username_entry.get()
6     email = email_entry.get()
7     password = password_entry.get()
8
9     try:
10         # 1. Validate empty fields
11         if username == '' or email == '' or password == '':
12             raise ValueError("All fields must be filled.")
13
14         # 2. Validate email format using regex
15         email_pattern = r'[^@]+@[^@]+\.[^@]+'
16         if not re.match(email_pattern, email):
17             raise ValueError("Invalid email format.")
18
19         # If all validations pass:
20         result.config(text="Form submitted successfully!", fg="green")
21
22     except ValueError as error:
23         result.config(text=str(error), fg="red")
24
25
26 root = Tk()
27 root.title("User Entry Form")
28
29 # --- Username ---
30 Label(root, text="Username:").pack()
31 username_entry = Entry(root, width=30)
32 username_entry.pack()
33
34 # --- Email ---
35 Label(root, text="Email:").pack()
36 email_entry = Entry(root, width=30)
37 email_entry.pack()
38
39 # --- Password ---
40 Label(root, text="Password:").pack()
41 password_entry = Entry(root, width=30, show="**")
42 password_entry.pack()
43
44 # --- Submit Button ---
45 Button(root, text="Submit", command=submit_form).pack(pady=10)
46
47 # --- Result Message ---
48 result = Label(root, text="")
49 result.pack()
50
51 root.mainloop()
```

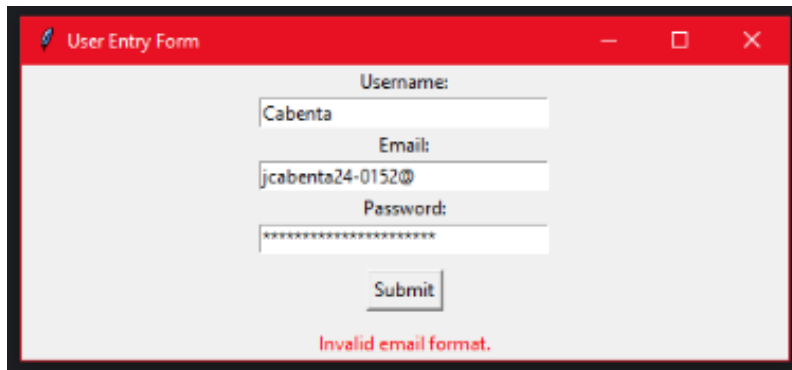
### III. SAMPLE OUTPUT



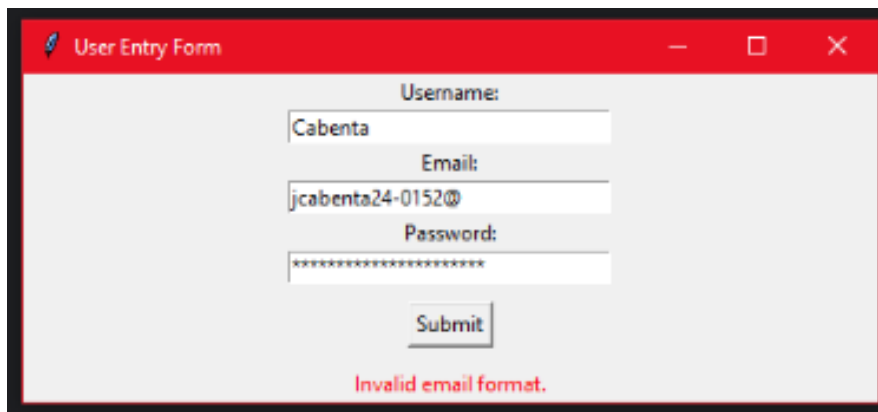
A screenshot of a Windows application window titled "User Entry Form". The window has a red title bar. Inside, there are three text input fields labeled "Username:", "Email:", and "Password:". The "Username" field contains "Cabenta", the "Email" field contains "jcabenta24-0152@cca.edu.ph", and the "Password" field contains a series of asterisks. Below the fields is a "Submit" button. At the bottom of the window, a green message reads "Form submitted successfully!".



A screenshot of the "User Entry Form" window. The "Username" field is empty, while the "Email" and "Password" fields contain "jcabenta24-0152@cca.edu.ph" and asterisks respectively. The "Submit" button is visible. At the bottom, a red message reads "All fields must be filled."



A screenshot of the "User Entry Form" window. The "Username" field contains "Cabenta", the "Email" field contains "jcabenta24-0152@", and the "Password" field contains asterisks. The "Submit" button is visible. At the bottom, a red message reads "Invalid email format."



A screenshot of the "User Entry Form" window. The "Username" field contains "Cabenta", the "Email" field contains "jcabenta24-0152@", and the "Password" field contains asterisks. The "Submit" button is visible. At the bottom, a red message reads "Invalid email format."