MIT World Peace University

Python Programming
Second Year B. Tech, Semester 4

Learning Basics of the *Tkinter Module*

Assignment No. 8

Prepared By

Krishnaraj Thadesar
Cyber Security and Forensics
Batch A1, PA 20

April 30, 2023

Contents

1 Aim **1**

2 Objectives **1**

3 Problem Statement **1**

4 Theory **1**

5 Input and Output **2**

5.1 Input 2

5.2 Output 2

6 Requirements **4**

7 Code **4**

7.1 Basic tkinter gui 4

7.2 Assignment 5

8 Conclusion **7**

9 FAQ **8**

1 Aim

Write a Python GUI program to import Tkinter package and create a program to take input of your date of output your birth and age.

2 Objectives

1. To learn and implement Python GUI using Tkinter.

3 Problem Statement

Create a program to take input of your date of output your birth and age.

4 Theory

Tkinter commonly comes bundled with Python, using Tk and is Python's standard GUI framework. It is famous for its simplicity and graphical user interface. It is open-source and available under the Python License.

Note: Tkinter comes pre-installed with Python 3 , and you need not bother about installing it. Now, let's build a very simple GUI with the help of Tkinter and understand it with the help of a flow diagram.

Steps:

1. First, you import the key component, i.e., the Tkinter module.
2. As a next step, you initialize the window manager with the tkinter.Tk() method and assign it to a variable. This method creates a blank window with close, maximize, and minimize buttons on the top as a usual GUI should have.
3. Then as an optional step, you will Rename the title of the window as you like with window.title(title).
4. Next, you make use of a widget called Label, which is used to insert some text into the window.
5. Then, you make use of Tkinter's geometry management attribute called pack() to display the widget in size it requires.
6. Finally, as the last step, you use the mainloop() method to display the window until you manually close it. It runs an infinite loop in the backend. Widgets

- Button: Button widget has a property for switching on/off. When a user clicks the button, an event is triggered in the Tkinter. Syntax: buttonwidget = tk.Button(widget, option=placeholder) where widget is the argument for the parent window/frame while option is a placeholder that can have various values like foreground and background color, font, command (for function call), image, height, and width of button.
- Canvas: Canvas is used to draw shapes in your GUI and supports various drawing methods. Syntax: canvaswidget = tk.Canvas (widget, option = placeholder) where widget is the param-

eter for the parent window/frame while option is a placeholder that can have various values like border-width, background color, height and width of widget.

- Checkbutton: Checkbutton records on-off or true-false state. It lets you can select more than one option at a time and even leave it unchecked. Syntax: checkbuttonwidget = tk.CheckButton(widget, option=placeholder) where widget is the parameter for the parent window/frame while option is a placeholder that can have various values like title, text, background and foreground color while widget is under the cursor, font, image, etc.
- Entry: Entry widget is used to create input fields or to get input text from the user within the GUI. Syntax: entrywidget = tk.Entry (widget, option=placeholder) where widget is the parameter for the parent window/frame while option is a placeholder that can have various values like border-width, background color, width and height of button etc.
- Frame: Frame is used as containers in the Tkinter for grouping and adequately organizing the widgets. Syntax: framewidget = tk.Frame(widget, option=placeholder) where widget is the parameter for the parent window/frame while option is a placeholder that can have various values like border-width, height and width of widget, highlightcolor (color when widget has to be focused).
- Label: Label is used to create a single line widgets like text, images, etc. Syntax: labelwidget = tk.Label(widget, option=placeholder) where widget is the parameter for the parent window or frame while option is a placeholder that can have various values like the font of a button, background color, image, width, and height of button.

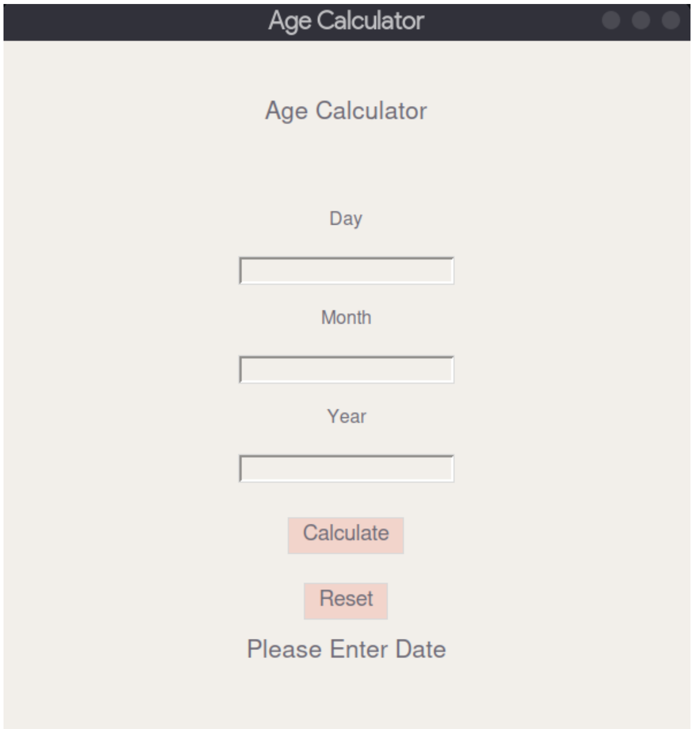
5 Input and Output

5.1 Input

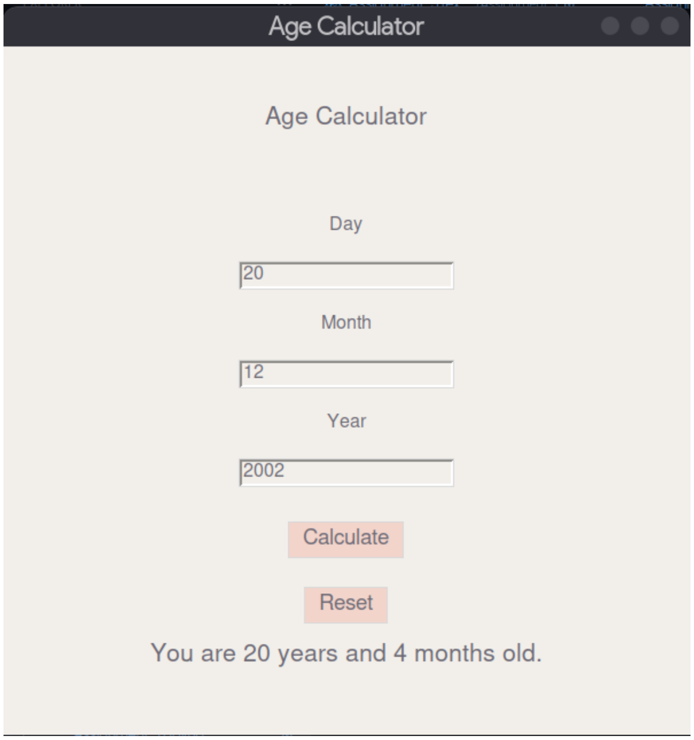
Tkinter GUI with input as date of birth of a student in date format.

5.2 Output

Tkinter GUI with input as date of birth of a student and output shows age.



Please Enter Date

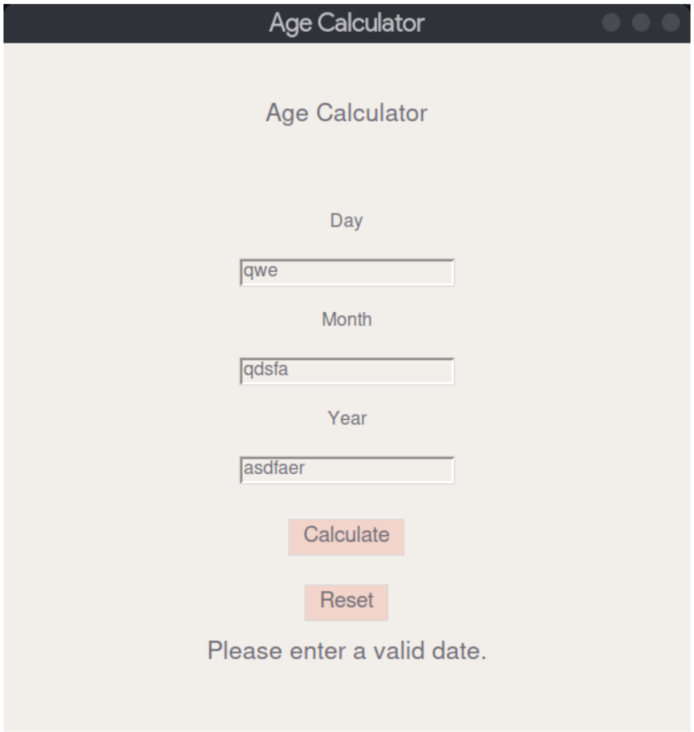


Day: 20

Month: 12

Year: 2002

You are 20 years and 4 months old.



Age Calculator

Day: qwe

Month: qdsfa

Year: asdfaer

Please enter a valid date.

6 Requirements

1. Python 3.7 or above
2. tkinter module comes inbuilt with python

7 Code

7.1 Basic tkinter gui

[24]:

```
from tkinter import *

# size of the window is defined
root = Tk()
root.geometry("400x400")

frame_1 = Frame(root)
frame_1.pack(side=TOP)
frame_2 = Frame(root)
frame_2.pack(side=BOTTOM)


button_1 = Button(frame_1, text="Button 1", fg="red")
button_2 = Button(frame_1, text="Button 2", fg="blue")

# to add the button the respective screens
```

```
button_1.pack()
button_2.pack()

root.mainloop()
```

7.2 Assignment

[25]:

```
import datetime
import tkinter as tk

# Define the pastel color scheme
bg_color = "#F2EFEA"
fg_color = "#6D6A75"
button_color = "#F2D4CB"
button_hover_color = "#F2D4CB"

def calculate_age():
    # get the current date
    try:
        today = datetime.date.today()
        # get the date of birth from the input fields
        day = int(day_entry.get())
        month = int(month_entry.get())
        year = int(year_entry.get())
        print(day, month, year)
        dob = datetime.date(year, month, day)
        # calculate the difference between the current date and date of birth
        age = today - dob
        # calculate age in years and months
        years = int(age.days / 365.25)
        months = int((age.days % 365.25) / 30.44)
        # update the label with the age
        age_label.config(text="You are {} years and {} months old.".
 ↪format(years, months))
    except:
        age_label.config(text="Please enter a valid date.")

def reset_fields():
    # clear the input fields and age label
    day_entry.delete(0, tk.END)
    month_entry.delete(0, tk.END)
    year_entry.delete(0, tk.END)
    age_label.config(text="")

# Create a new tkinter window
root = tk.Tk()
root.title("Age Calculator")
root.geometry("600x600")
root.configure(bg=bg_color)

main_label = tk.Label(root, text="Age Calculator", bg=bg_color, fg=fg_color,␣
 ↪font=("Helvetica", 16))
main_label.pack(pady=(50, 20))

# Create the input fields for date of birth
day_label = tk.Label(root, text="Day", bg=bg_color, fg=fg_color,␣
 ↪font=("Helvetica", 12))
day_entry = tk.Entry(root, font=("Helvetica", 12), bg=bg_color, fg=fg_color,␣
 ↪bd=2)
month_label = tk.Label(root, text="Month", bg=bg_color, fg=fg_color,␣
 ↪font=("Helvetica", 12))
month_entry = tk.Entry(root, font=("Helvetica", 12), bg=bg_color, fg=fg_color,␣
 ↪bd=2)
year_label = tk.Label(root, text="Year", bg=bg_color, fg=fg_color,␣
 ↪font=("Helvetica", 12))
year_entry = tk.Entry(root, font=("Helvetica", 12), bg=bg_color, fg=fg_color,␣
 ↪bd=2)

# Create the button to calculate age
calc_button = tk.Button(root, text="Calculate", bg=button_color, fg=fg_color,␣
 ↪font=("Helvetica", 14), bd=0,
                        activebackground=button_hover_color,␣
 ↪activeforeground=fg_color, command=calculate_age)


# Create the button to reset the fields
reset_button = tk.Button(root, text="Reset", bg=button_color, fg=fg_color,␣
 ↪font=("Helvetica", 14), bd=0,
                         activebackground=button_hover_color,␣
 ↪activeforeground=fg_color, command=reset_fields)


# Create the label to display the age
age_label = tk.Label(root, text="Please Enter Date", bg=bg_color, fg=fg_color,␣
 ↪font=("Helvetica", 16))

# Add the widgets to the window
day_label.pack(pady=(50, 10))
day_entry.pack(pady=10)
month_label.pack(pady=10)
month_entry.pack(pady=10)
year_label.pack(pady=10)
year_entry.pack(pady=10)
```

```
calc_button.pack(pady=20)
reset_button.pack(pady=5)
age_label.pack(pady=10)

# Run the main tkinter event loop
root.mainloop()
```

```
20 12 2002
```

[ ]:

```

```

8 Conclusion

Studied Python GUI using Tkinter with various widgets.

9 FAQ

1. **What is python Tkinter?**

   Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI. It is a thin object-oriented layer on top of Tcl/Tk. Tkinter is not the only GuiProgramming toolkit for Python. It is however the most commonly used one, and almost the only one that is portable between Unix, Mac and Windows. It was originally written for Python 1.4, and is now the standard way to create a GUI with Python.

   There are also other GUI toolkits like wxPython, PyQt, Kivy, etc.

2. **What is Python Tkinter pack() and mainloop() method?**

   pack() method: This geometry manager organizes widgets in blocks before placing them in the parent widget.

   mainloop() method: There is a method known by the name mainloop() is used when your application is ready to run. mainloop() is an infinite loop used to run the application, wait for an event to occur and process the event as long as the window is not closed.

```python
# Python program to create a simple GUI and show an example of pack() method and mainloop() method

# Importing Tkinter module
from tkinter import *

root = Tk()

# Creating a Label Widget
myLabel1 = Label(root, text="Hello World!")

# Showing it onto the screen
myLabel1.pack()

# Creating a Label Widget
myLabel2 = Label(root, text="My name is John Elder")

# Showing it onto the screen
myLabel2.pack()

root.mainloop() // runs in a infinite loop
```