POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

Sta. Mesa Manila

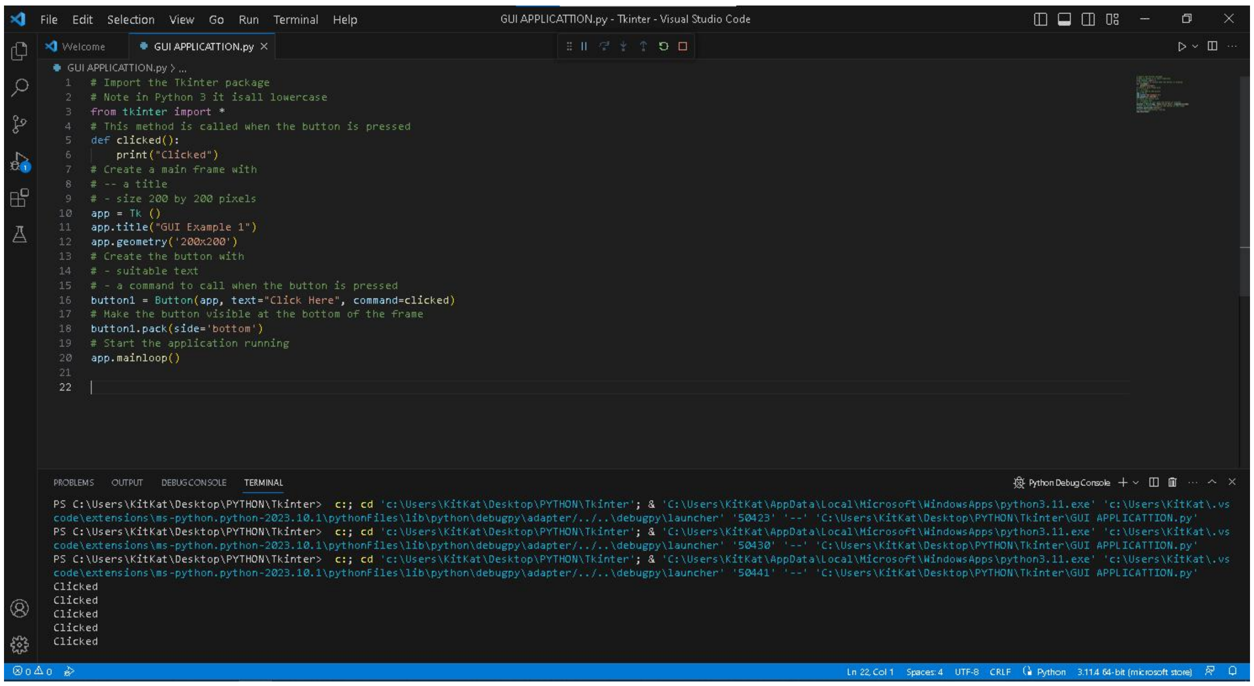
College of Engineering
Department of Computer Engineering

Name: Katrina Ricci C. Batin **Course/Year/Section:** BSCPE1-2
Date: June 10, 2023
Subject: Object Oriented Programming
Prof. Eng. Julius Cansino

Introduction to GUI Programming in Python

LEX 6.0 TKinter -GUI Application in Python

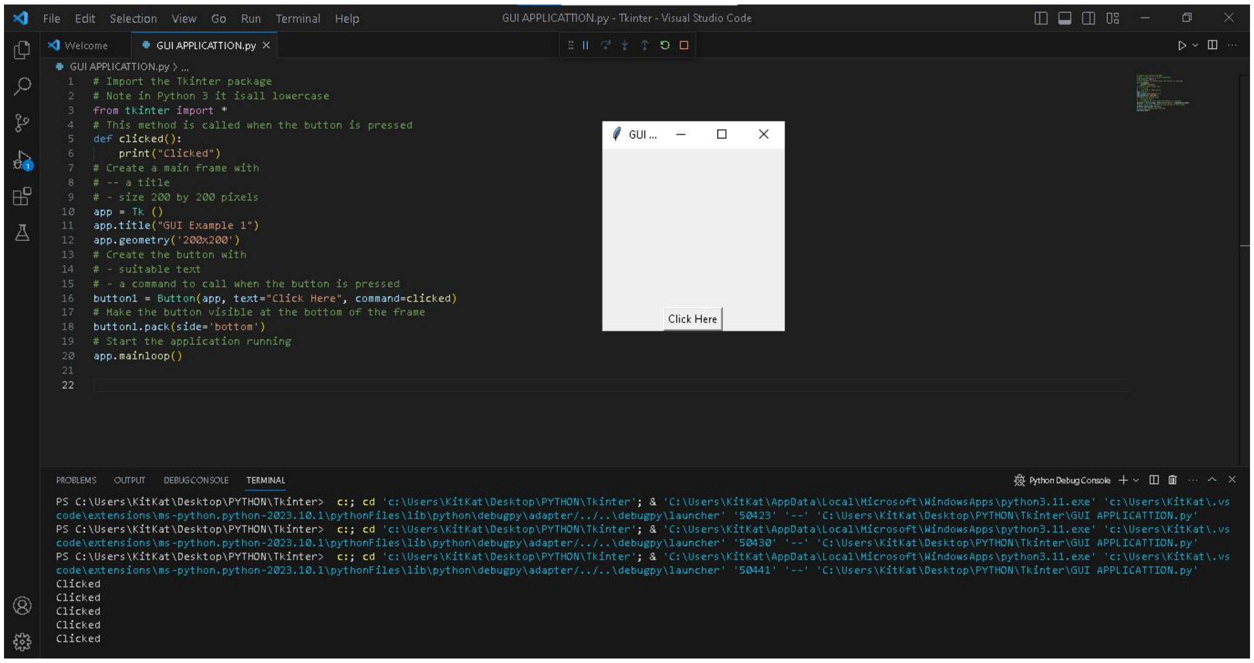
Exercise 1: A First GUI Program The following program is available for download (called exercise1.py). Find the program, open it using IDLE and run it.


File Edit Selection View Go Run Terminal Help
GUI APPLICATTION.py - Tkinter - Visual Studio Code
Welcome
GUI APPLICATTION.py
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python Debug Console
Ln 22, Col 1 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit (microsoft store)

```python
# Import the Tkinter package
# Note in Python 3 it isall lowercase
from tkinter import *
# This method is called when the button is pressed
def clicked():
    print("Clicked")
# Create a main frame with
# -- a title
# - size 200 by 200 pixels
app = Tk ()
app.title("GUI Example 1")
app.geometry('200x200')
# Create the button with
# - suitable text
# - a command to call when the button is pressed
button1 = Button(app, text="Click Here", command=clicked)
# Make the button visible at the bottom of the frame
button1.pack(side='bottom')
# Start the application running
app.mainloop()

```

```
PS C:\Users\KitKat\Desktop\PYTHON\Tkinter>  c:; cd 'c:\Users\KitKat\Desktop\PYTHON\Tkinter'; & 'C:\Users\KitKat\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\KitKat\.vs
code\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '50423' '--' 'C:\Users\KitKat\Desktop\PYTHON\Tkinter\GUI APPLICATTION.py'
PS C:\Users\KitKat\Desktop\PYTHON\Tkinter>  c:; cd 'c:\Users\KitKat\Desktop\PYTHON\Tkinter'; & 'C:\Users\KitKat\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\KitKat\.vs
code\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '50430' '--' 'C:\Users\KitKat\Desktop\PYTHON\Tkinter\GUI APPLICATTION.py'
PS C:\Users\KitKat\Desktop\PYTHON\Tkinter>  c:; cd 'c:\Users\KitKat\Desktop\PYTHON\Tkinter'; & 'C:\Users\KitKat\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\KitKat\.vs
code\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '50441' '--' 'C:\Users\KitKat\Desktop\PYTHON\Tkinter\GUI APPLICATTION.py'
Clicked
Clicked
Clicked
Clicked
Clicked
```

OUTPUT:


File Edit Selection View Go Run Terminal Help
GUI APPLICATTION.py - Tkinter - Visual Studio Code
Welcome
GUI APPLICATTION.py
GUI ...
Click Here
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python Debug Console

```python
# Import the Tkinter package
# Note in Python 3 it isall lowercase
from tkinter import *
# This method is called when the button is pressed
def clicked():
    print("Clicked")
# Create a main frame with
# -- a title
# - size 200 by 200 pixels
app = Tk ()
app.title("GUI Example 1")
app.geometry('200x200')
# Create the button with
# - suitable text
# - a command to call when the button is pressed
button1 = Button(app, text="Click Here", command=clicked)
# Make the button visible at the bottom of the frame
button1.pack(side='bottom')
# Start the application running
app.mainloop()
```

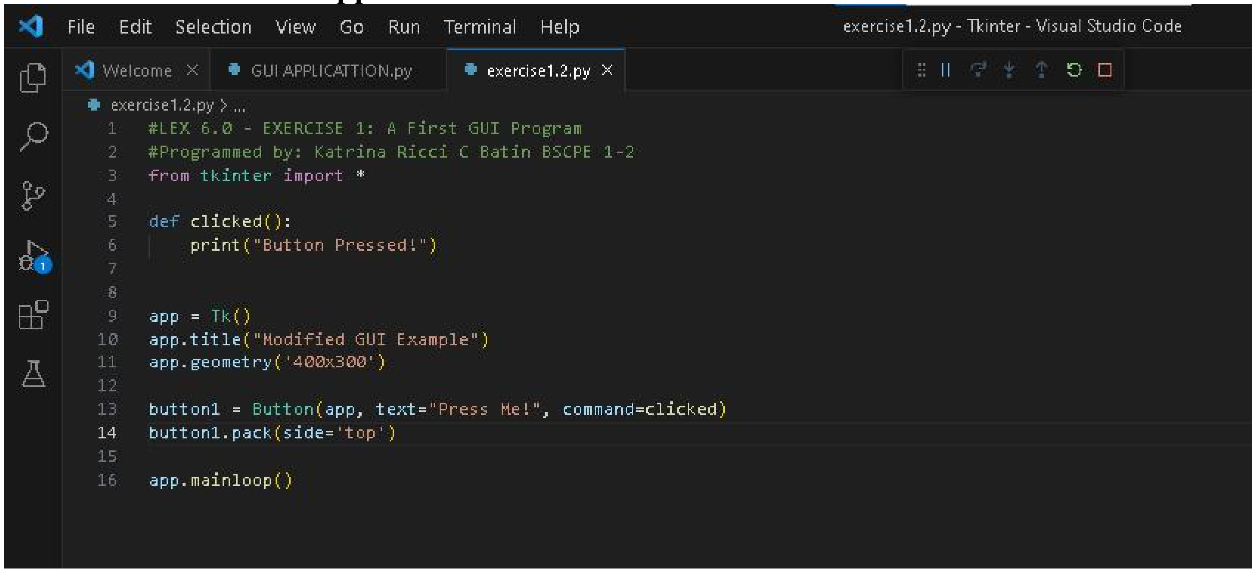
```
PS C:\Users\KitKat\Desktop\PYTHON\Tkinter>  c:; cd 'c:\Users\KitKat\Desktop\PYTHON\Tkinter'; & 'C:\Users\KitKat\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\KitKat\.vscode\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '50423' '--' 'C:\Users\KitKat\Desktop\PYTHON\Tkinter\GUI APPLICATTION.py'
PS C:\Users\KitKat\Desktop\PYTHON\Tkinter>  c:; cd 'c:\Users\KitKat\Desktop\PYTHON\Tkinter'; & 'C:\Users\KitKat\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\KitKat\.vscode\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '50430' '--' 'C:\Users\KitKat\Desktop\PYTHON\Tkinter\GUI APPLICATTION.py'
PS C:\Users\KitKat\Desktop\PYTHON\Tkinter>  c:; cd 'c:\Users\KitKat\Desktop\PYTHON\Tkinter'; & 'C:\Users\KitKat\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\KitKat\.vscode\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '50441' '--' 'C:\Users\KitKat\Desktop\PYTHON\Tkinter\GUI APPLICATTION.py'
Clicked
Clicked
Clicked
Clicked
Clicked
```

Exercise 1.2: Modify the Program

Althoughithasnotbeenexplainedyet,seeifyoucanfigureouthowtomake the followingmodifications:

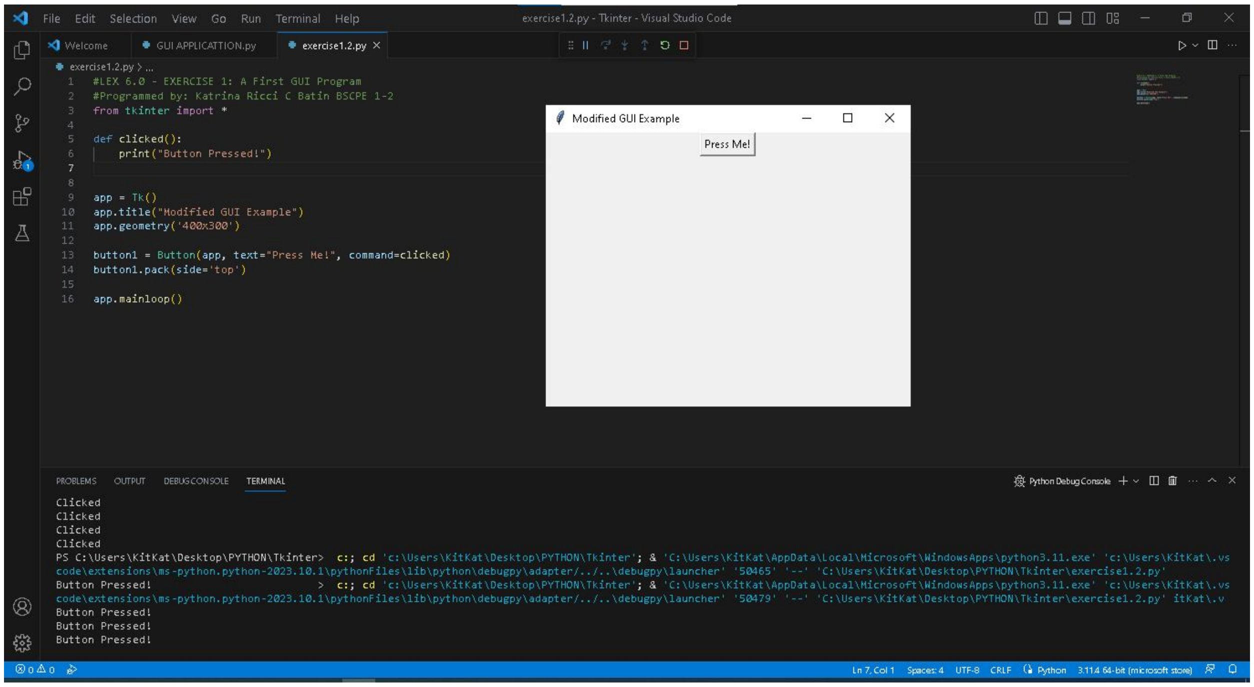
- Change thetitle
- Change the text in the button
- Change the text printed when the button is pressed
- Change the size (geometry) of the rectangular frame
- Move the button to the top of the frame

Code: Modified GUI Application


File Edit Selection View Go Run Terminal Help
exercise1.2.py - Tkinter - Visual Studio Code
Welcome
GUI APPLICATTION.py
exercise1.2.py
exercise1.2.py > ...

```python
#LEX 6.0 - EXERCISE 1: A First GUI Program
#Programmed by: Katrina Ricci C Batin BSCPE 1-2
from tkinter import *

def clicked():
    print("Button Pressed!")


app = Tk()
app.title("Modified GUI Example")
app.geometry('400x300')

button1 = Button(app, text="Press Me!", command=clicked)
button1.pack(side='top')

app.mainloop()
```

OUTPUT:



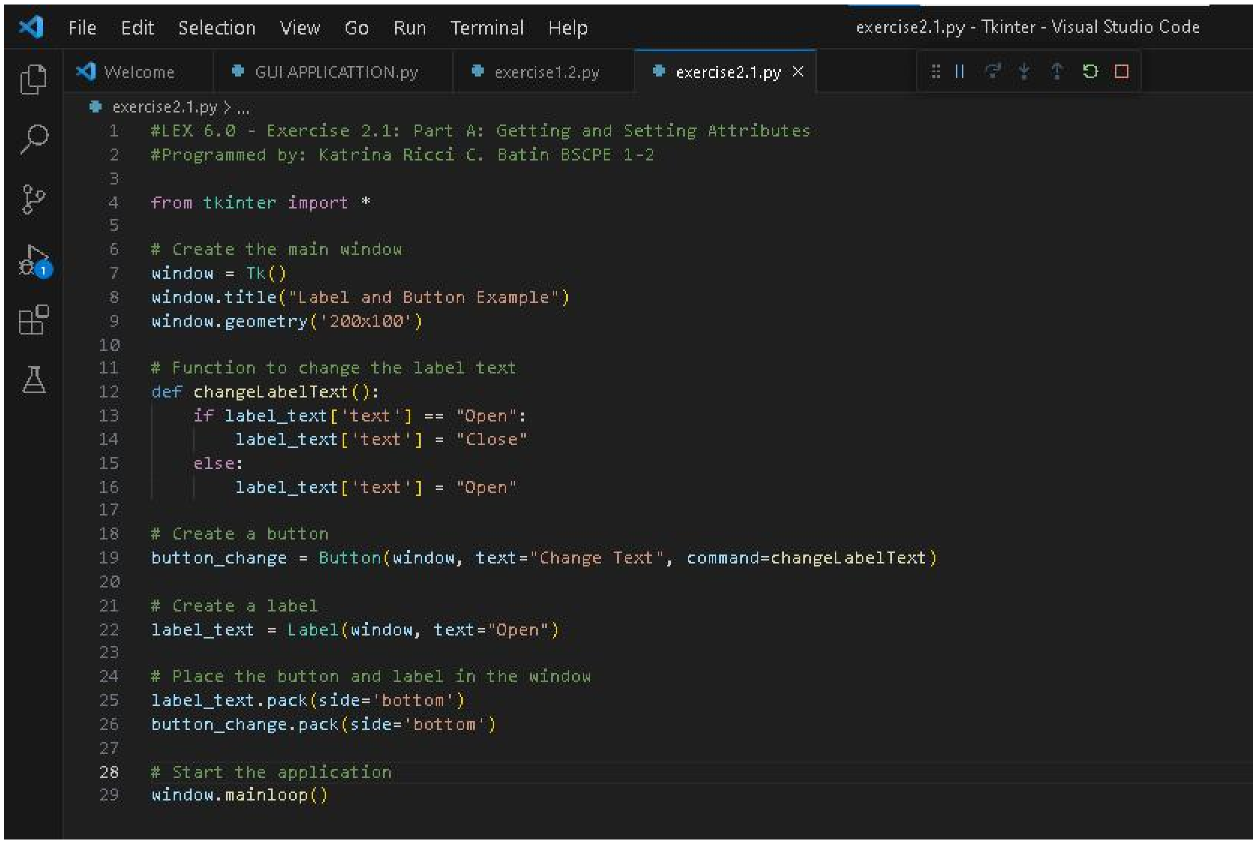
```
Clicked
Clicked
Clicked
Clicked
PS C:\Users\KitKat\Desktop\PYTHON\Tkinter>  c:; cd 'c:\Users\KitKat\Desktop\PYTHON\Tkinter'; & 'C:\Users\KitKat\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\KitKat\.vs
code\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '50465' '--' 'C:\Users\KitKat\Desktop\PYTHON\Tkinter\exercise1.2.py'
Button Pressed!                          >  c:; cd 'c:\Users\KitKat\Desktop\PYTHON\Tkinter'; & 'C:\Users\KitKat\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\KitKat\.vs
code\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '50479' '--' 'C:\Users\KitKat\Desktop\PYTHON\Tkinter\exercise1.2.py' itKat\.v
Button Pressed!
Button Pressed!
Button Pressed!
```

Exercise 2: Adding a Label and Entry Widget

Exercise2.1:PartA:GettingandSettingAttributes

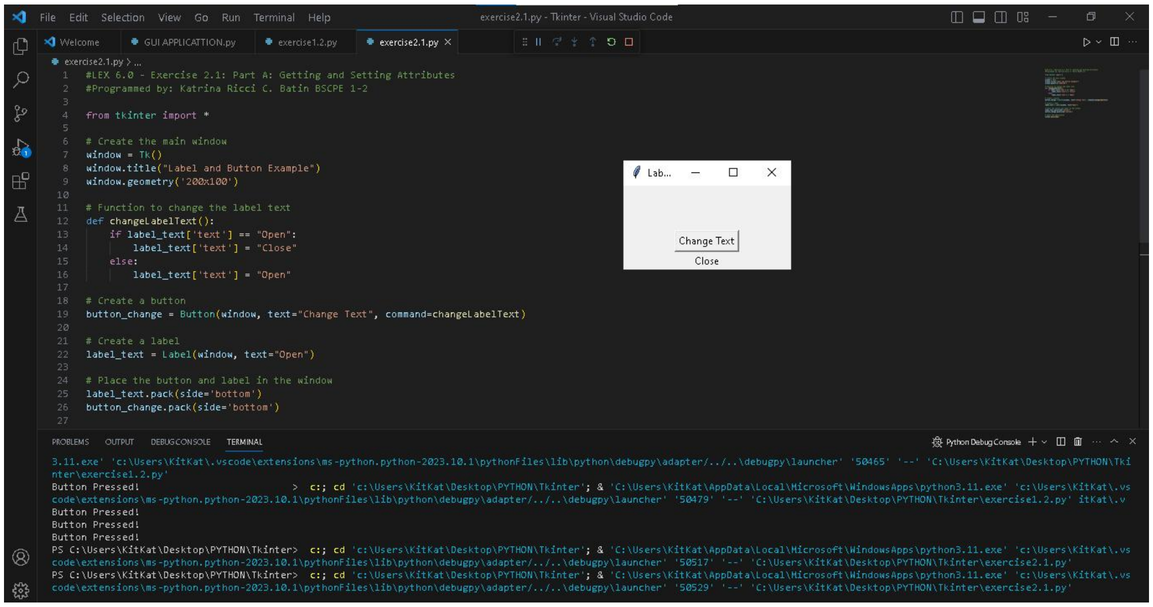
Run the given program (exercise2A.py); this version just has a button and a label. Pressing the button once changes the text of the label. Change it so that the text changes on each press, toggling between two messages.

Code: Part A: Getting and Setting Attributes



```python
#LEX 6.0 - Exercise 2.1: Part A: Getting and Setting Attributes
#Programmed by: Katrina Ricci C. Batin BSCPE 1-2

from tkinter import *

# Create the main window
window = Tk()
window.title("Label and Button Example")
window.geometry('200x100')

# Function to change the label text
def changeLabelText():
    if label_text['text'] == "Open":
        label_text['text'] = "Close"
    else:
        label_text['text'] = "Open"

# Create a button
button_change = Button(window, text="Change Text", command=changeLabelText)

# Create a label
label_text = Label(window, text="Open")

# Place the button and label in the window
label_text.pack(side='bottom')
button_change.pack(side='bottom')

# Start the application
window.mainloop()
```

OUTPUTT

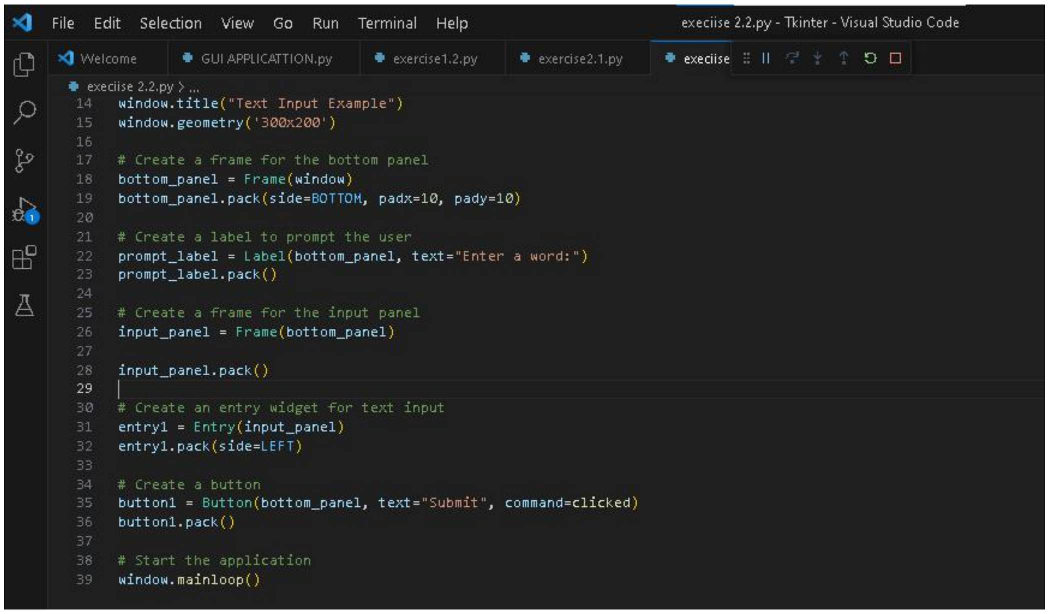


```
3.11.exe' 'c:\Users\KitKat\.vscode\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '50465' '--' 'C:\Users\KitKat\Desktop\PYTHON\Tkinter\exercise1.2.py'
Button Pressed!                    >  c:; cd 'c:\Users\KitKat\Desktop\PYTHON\Tkinter'; & 'C:\Users\KitKat\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\KitKat\.vscode\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '50479' '--' 'C:\Users\KitKat\Desktop\PYTHON\Tkinter\exercise1.2.py' itKat\.v
Button Pressed!
Button Pressed!
Button Pressed!
PS C:\Users\KitKat\Desktop\PYTHON\Tkinter>  c:; cd 'c:\Users\KitKat\Desktop\PYTHON\Tkinter'; & 'C:\Users\KitKat\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\KitKat\.vscode\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '50517' '--' 'C:\Users\KitKat\Desktop\PYTHON\Tkinter\exercise2.1.py'
PS C:\Users\KitKat\Desktop\PYTHON\Tkinter>  c:; cd 'c:\Users\KitKat\Desktop\PYTHON\Tkinter'; & 'C:\Users\KitKat\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\KitKat\.vscode\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '50529' '--' 'C:\Users\KitKat\Desktop\PYTHON\Tkinter\exercise2.1.py'
```

Exercise 2.2: Part B: Complete Program

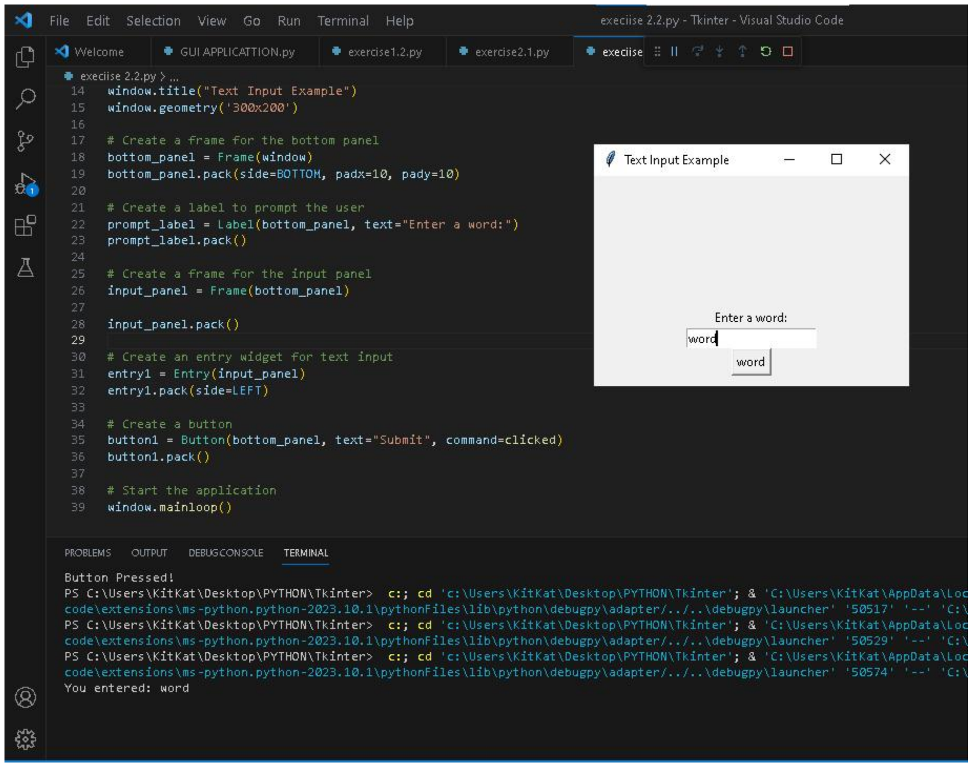
Run the given program (exercise2B.py); this part adds the entry widget. When you enter text in the box (the Entry widget) and press the button, it only prints the text from the entry. Complete it so that it behaves as described above.

Code: Part B: Complete Program



```python
window.title("Text Input Example")
window.geometry('300x200')

# Create a frame for the bottom panel
bottom_panel = Frame(window)
bottom_panel.pack(side=BOTTOM, padx=10, pady=10)

# Create a label to prompt the user
prompt_label = Label(bottom_panel, text="Enter a word:")
prompt_label.pack()

# Create a frame for the input panel
input_panel = Frame(bottom_panel)

input_panel.pack()

# Create an entry widget for text input
entry1 = Entry(input_panel)
entry1.pack(side=LEFT)

# Create a button
button1 = Button(bottom_panel, text="Submit", command=clicked)
button1.pack()

# Start the application
window.mainloop()
```

OUTPUT :



```python
window.title("Text Input Example")
window.geometry('300x200')

# Create a frame for the bottom panel
bottom_panel = Frame(window)
bottom_panel.pack(side=BOTTOM, padx=10, pady=10)

# Create a label to prompt the user
prompt_label = Label(bottom_panel, text="Enter a word:")
prompt_label.pack()

# Create a frame for the input panel
input_panel = Frame(bottom_panel)

input_panel.pack()

# Create an entry widget for text input
entry1 = Entry(input_panel)
entry1.pack(side=LEFT)

# Create a button
button1 = Button(bottom_panel, text="Submit", command=clicked)
button1.pack()

# Start the application
window.mainloop()
```

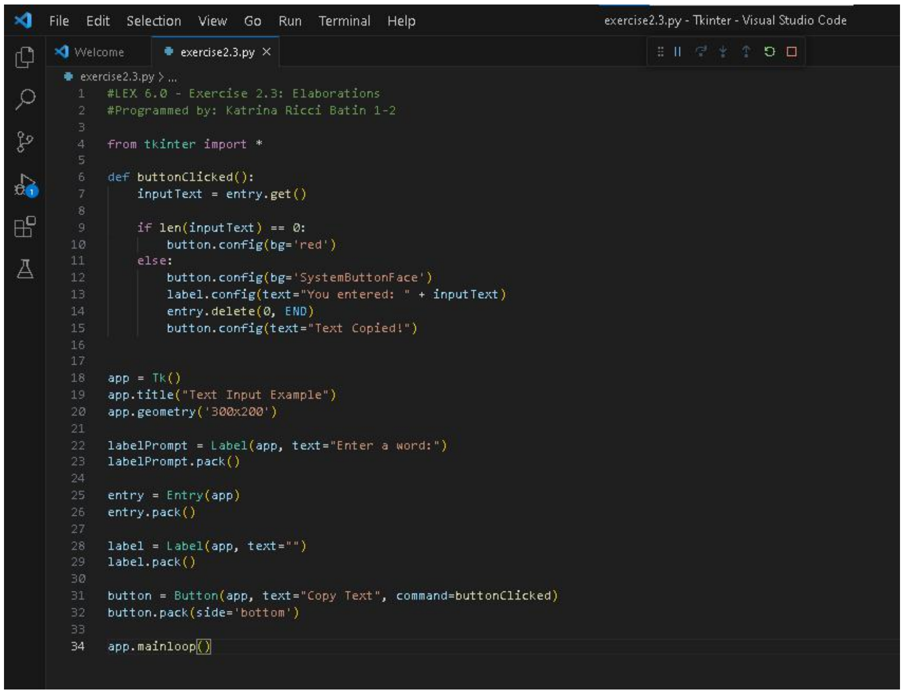
```
Button Pressed!
PS C:\Users\KitKat\Desktop\PYTHON\Tkinter>  c:; cd 'c:\Users\KitKat\Desktop\PYTHON\Tkinter'; & 'C:\Users\KitKat\AppData\Loc
code\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '50517' '--' 'C:\
PS C:\Users\KitKat\Desktop\PYTHON\Tkinter>  c:; cd 'c:\Users\KitKat\Desktop\PYTHON\Tkinter'; & 'C:\Users\KitKat\AppData\Loc
code\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '50529' '--' 'C:\
PS C:\Users\KitKat\Desktop\PYTHON\Tkinter>  c:; cd 'c:\Users\KitKat\Desktop\PYTHON\Tkinter'; & 'C:\Users\KitKat\AppData\Loc
code\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '50574' '--' 'C:\
You entered: word
```

Exercise 2.3:Elaborations

When the button is pressed, check if the entered text is blank (i.e. has zero length). If so, do not copy it but instead set the background of the button red. Restore the original background colour when the button is pressed and some text has been entered.

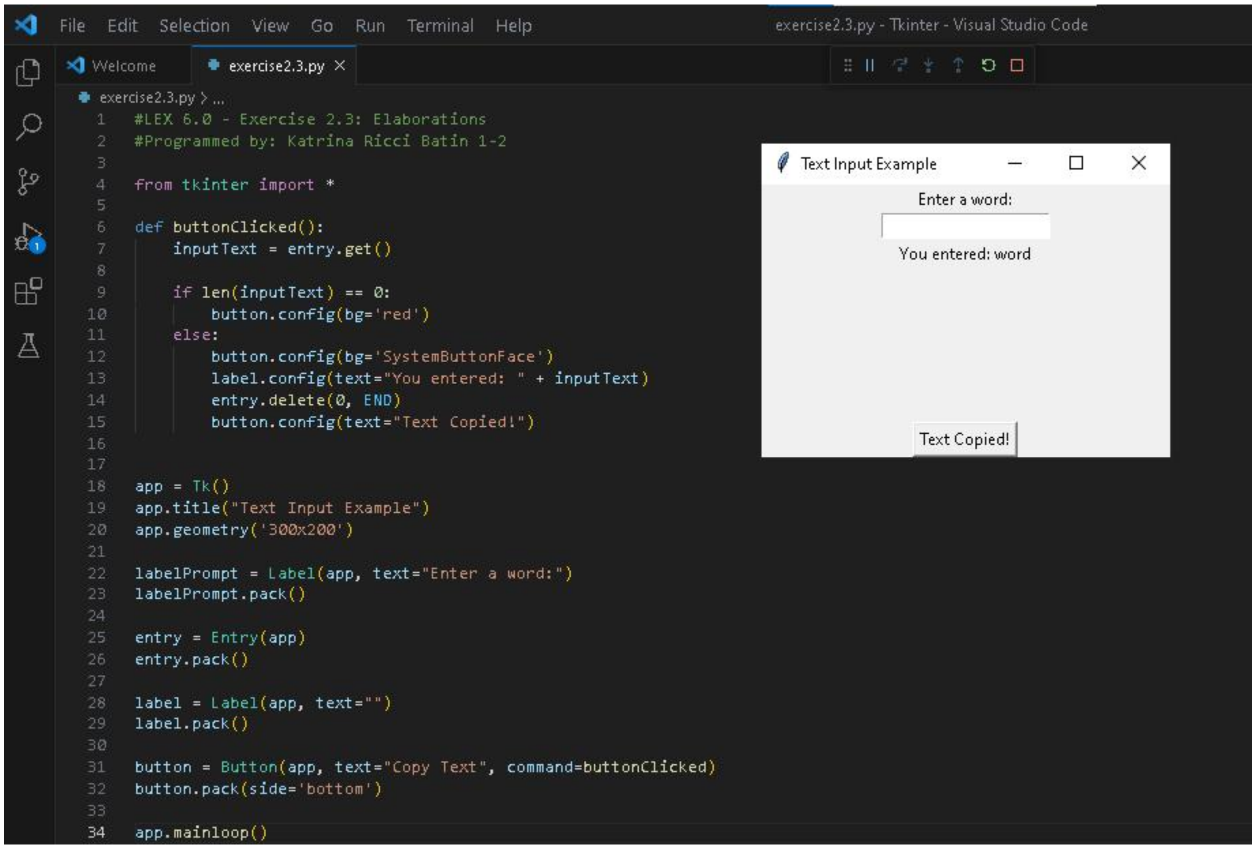
After the button has been pressed and the label changed, make the next press of the button clear the text in the entry widget. Change the button text so that the user understands what is happening.

Code:



```python
#LEX 6.0 - Exercise 2.3: Elaborations
#Programmed by: Katrina Ricci Batin 1-2

from tkinter import *

def buttonClicked():
    inputText = entry.get()

    if len(inputText) == 0:
        button.config(bg='red')
    else:
        button.config(bg='SystemButtonFace')
        label.config(text="You entered: " + inputText)
        entry.delete(0, END)
        button.config(text="Text Copied!")


app = Tk()
app.title("Text Input Example")
app.geometry('300x200')

labelPrompt = Label(app, text="Enter a word:")
labelPrompt.pack()

entry = Entry(app)
entry.pack()

label = Label(app, text="")
label.pack()

button = Button(app, text="Copy Text", command=buttonClicked)
button.pack(side='bottom')

app.mainloop()
```

OUTPUT :

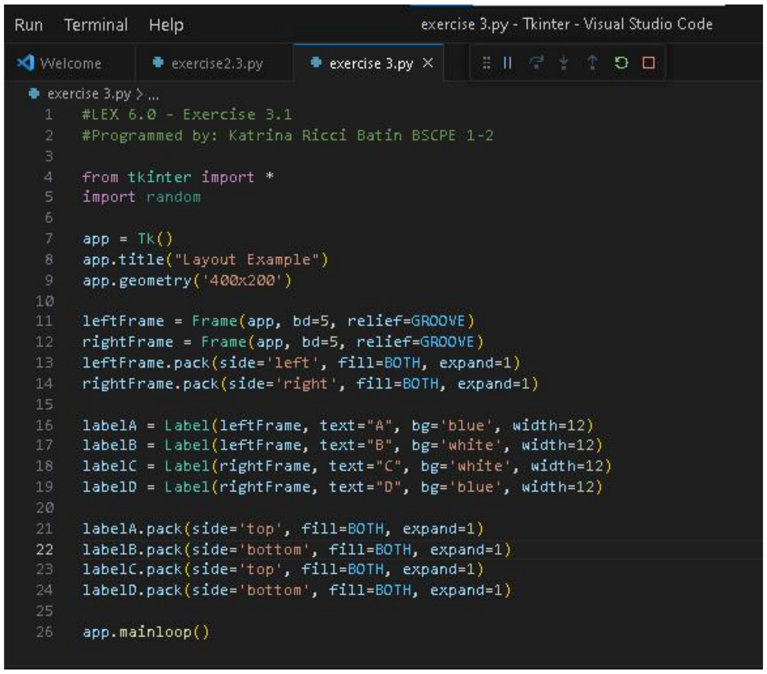


```python
#LEX 6.0 - Exercise 2.3: Elaborations
#Programmed by: Katrina Ricci Batin 1-2

from tkinter import *

def buttonClicked():
    inputText = entry.get()

    if len(inputText) == 0:
        button.config(bg='red')
    else:
        button.config(bg='SystemButtonFace')
        label.config(text="You entered: " + inputText)
        entry.delete(0, END)
        button.config(text="Text Copied!")


app = Tk()
app.title("Text Input Example")
app.geometry('300x200')

labelPrompt = Label(app, text="Enter a word:")
labelPrompt.pack()

entry = Entry(app)
entry.pack()

label = Label(app, text="")
label.pack()

button = Button(app, text="Copy Text", command=buttonClicked)
button.pack(side='bottom')

app.mainloop()
```

Exercise 3: Managing Layout

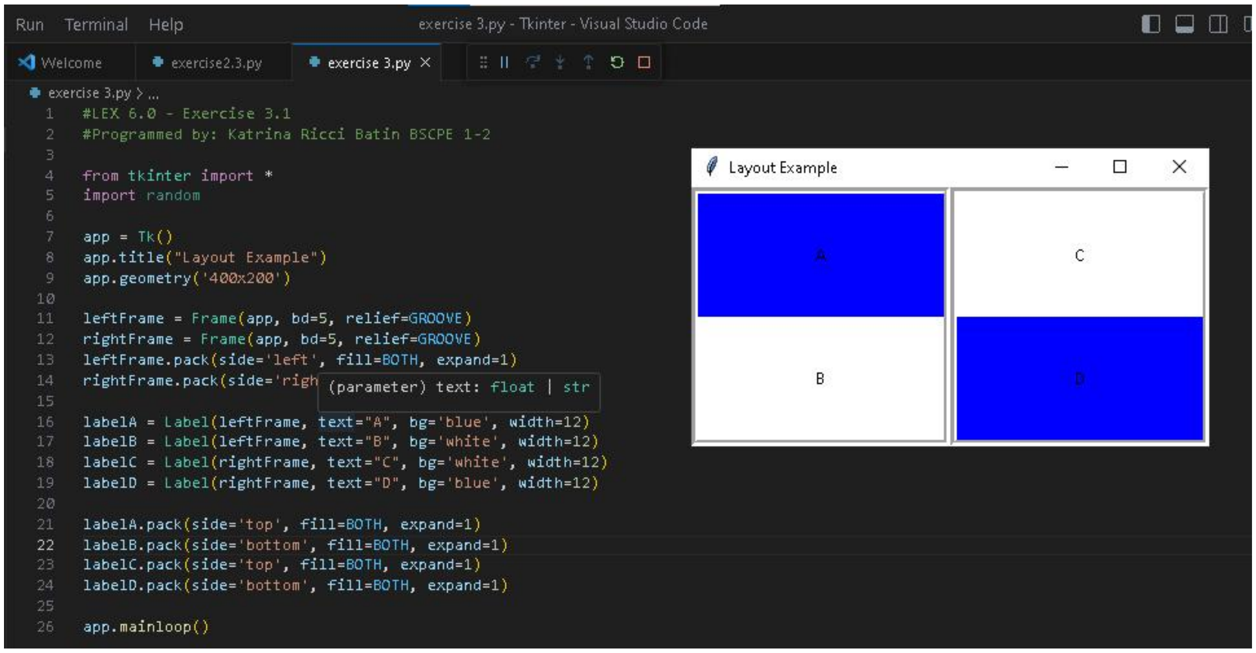
Exercise 3.1: Arrange the labels is a square grid with the pack layout manager this means introduces extra frames so that the labels are in the frames and the frames are in the top-level window. In the diagram above, the frames have a border so they can be seen.

Code:



```python
#LEX 6.0 - Exercise 3.1
#Programmed by: Katrina Ricci Batin BSCPE 1-2

from tkinter import *
import random

app = Tk()
app.title("Layout Example")
app.geometry('400x200')

leftFrame = Frame(app, bd=5, relief=GROOVE)
rightFrame = Frame(app, bd=5, relief=GROOVE)
leftFrame.pack(side='left', fill=BOTH, expand=1)
rightFrame.pack(side='right', fill=BOTH, expand=1)

labelA = Label(leftFrame, text="A", bg='blue', width=12)
labelB = Label(leftFrame, text="B", bg='white', width=12)
labelC = Label(rightFrame, text="C", bg='white', width=12)
labelD = Label(rightFrame, text="D", bg='blue', width=12)

labelA.pack(side='top', fill=BOTH, expand=1)
labelB.pack(side='bottom', fill=BOTH, expand=1)
labelC.pack(side='top', fill=BOTH, expand=1)
labelD.pack(side='bottom', fill=BOTH, expand=1)

app.mainloop()
```

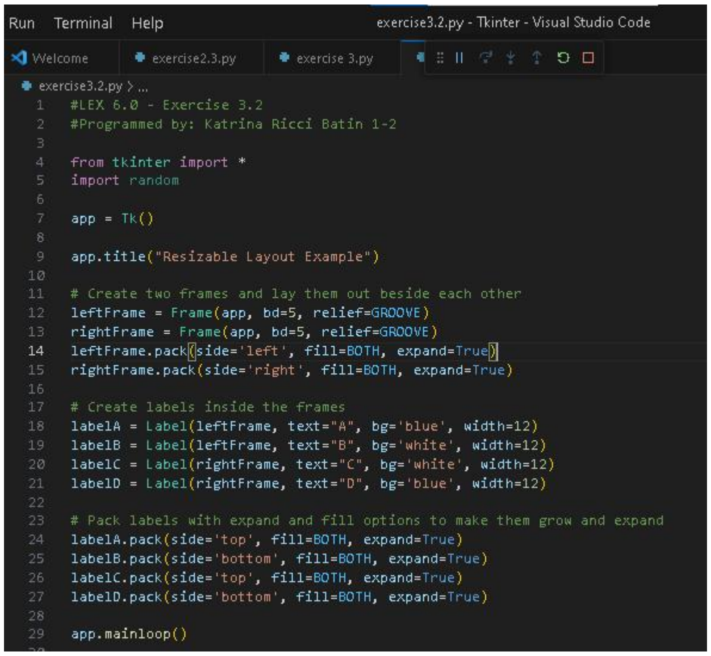
OUTPUT



```
#LEX 6.0 - Exercise 3.1
#Programmed by: Katrina Ricci Batin BSCPE 1-2

from tkinter import *
import random

app = Tk()
app.title("Layout Example")
app.geometry('400x200')

leftFrame = Frame(app, bd=5, relief=GROOVE)
rightFrame = Frame(app, bd=5, relief=GROOVE)
leftFrame.pack(side='left', fill=BOTH, expand=1)
rightFrame.pack(side='righ

labelA = Label(leftFrame, text="A", bg='blue', width=12)
labelB = Label(leftFrame, text="B", bg='white', width=12)
labelC = Label(rightFrame, text="C", bg='white', width=12)
labelD = Label(rightFrame, text="D", bg='blue', width=12)

labelA.pack(side='top', fill=BOTH, expand=1)
labelB.pack(side='bottom', fill=BOTH, expand=1)
labelC.pack(side='top', fill=BOTH, expand=1)
labelD.pack(side='bottom', fill=BOTH, expand=1)

app.mainloop()
```

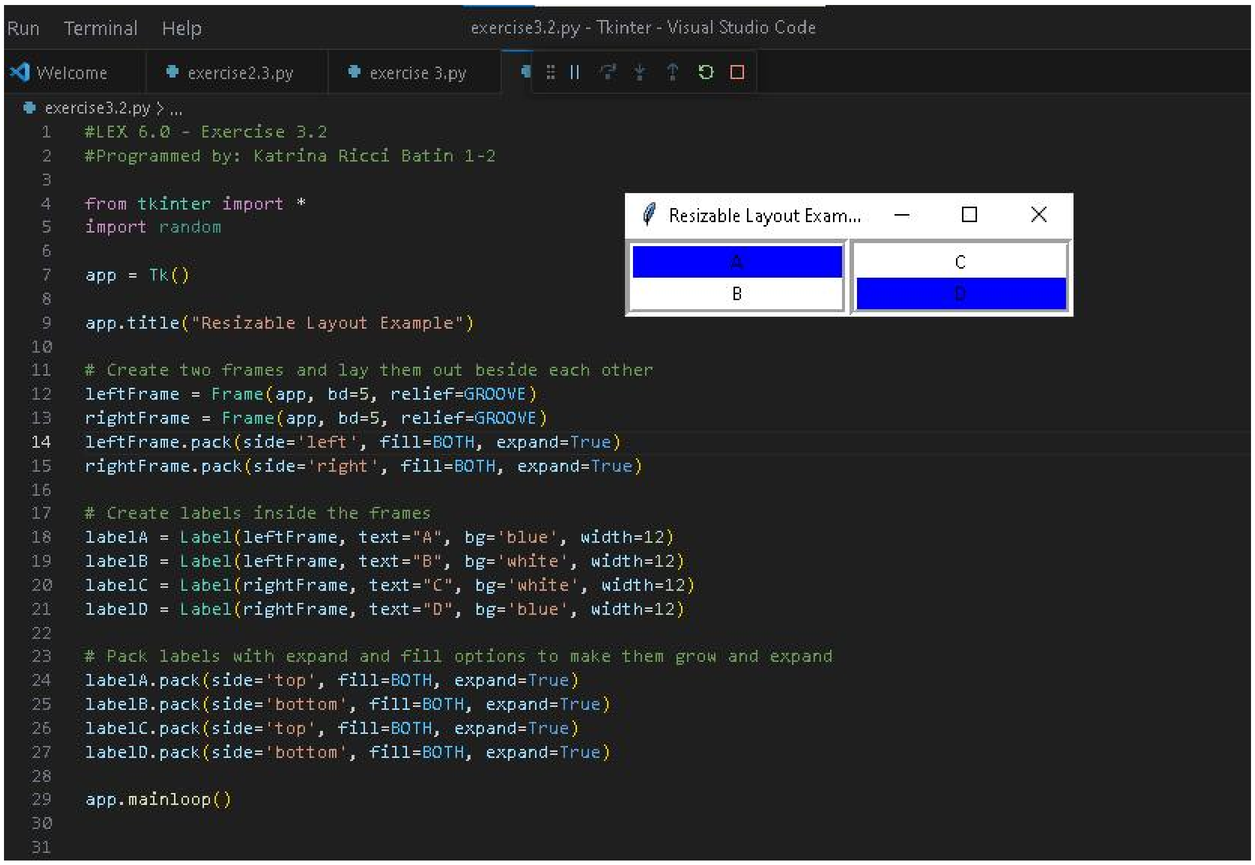
Exercise3.2: Support resizingUsethe'expand'and'fill'attributesofthepackmethodtomake the labels grow and expand into the available space. There is more guidance in code comments.

Code:



```
#LEX 6.0 - Exercise 3.2
#Programmed by: Katrina Ricci Batin 1-2

from tkinter import *
import random

app = Tk()

app.title("Resizable Layout Example")

# Create two frames and lay them out beside each other
leftFrame = Frame(app, bd=5, relief=GROOVE)
rightFrame = Frame(app, bd=5, relief=GROOVE)
leftFrame.pack(side='left', fill=BOTH, expand=True)
rightFrame.pack(side='right', fill=BOTH, expand=True)

# Create labels inside the frames
labelA = Label(leftFrame, text="A", bg='blue', width=12)
labelB = Label(leftFrame, text="B", bg='white', width=12)
labelC = Label(rightFrame, text="C", bg='white', width=12)
labelD = Label(rightFrame, text="D", bg='blue', width=12)

# Pack labels with expand and fill options to make them grow and expand
labelA.pack(side='top', fill=BOTH, expand=True)
labelB.pack(side='bottom', fill=BOTH, expand=True)
labelC.pack(side='top', fill=BOTH, expand=True)
labelD.pack(side='bottom', fill=BOTH, expand=True)

app.mainloop()
```

OUTPUT

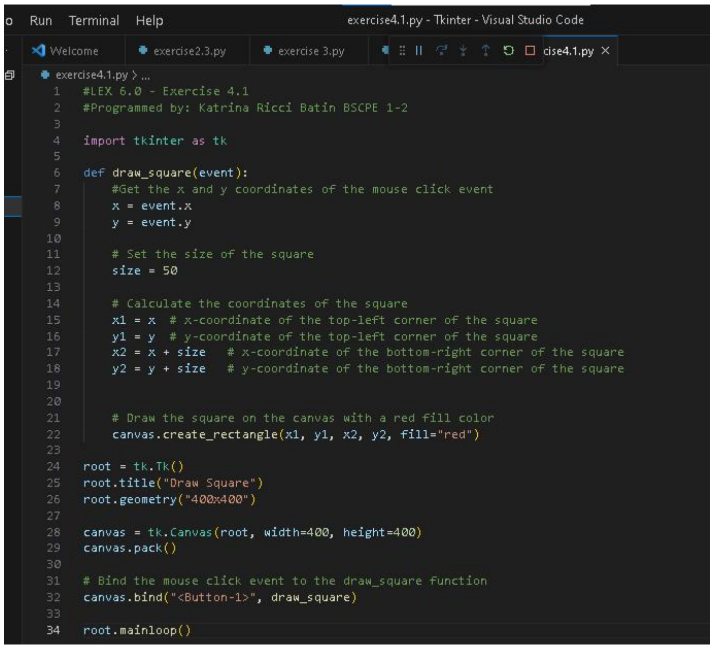


```
#LEX 6.0 - Exercise 3.2
#Programmed by: Katrina Ricci Batin 1-2

from tkinter import *
import random

app = Tk()

app.title("Resizable Layout Example")

# Create two frames and lay them out beside each other
leftFrame = Frame(app, bd=5, relief=GROOVE)
rightFrame = Frame(app, bd=5, relief=GROOVE)
leftFrame.pack(side='left', fill=BOTH, expand=True)
rightFrame.pack(side='right', fill=BOTH, expand=True)

# Create labels inside the frames
labelA = Label(leftFrame, text="A", bg='blue', width=12)
labelB = Label(leftFrame, text="B", bg='white', width=12)
labelC = Label(rightFrame, text="C", bg='white', width=12)
labelD = Label(rightFrame, text="D", bg='blue', width=12)

# Pack labels with expand and fill options to make them grow and expand
labelA.pack(side='top', fill=BOTH, expand=True)
labelB.pack(side='bottom', fill=BOTH, expand=True)
labelC.pack(side='top', fill=BOTH, expand=True)
labelD.pack(side='bottom', fill=BOTH, expand=True)

app.mainloop()
```

Exercise 4: The Drawing Canvas and Events

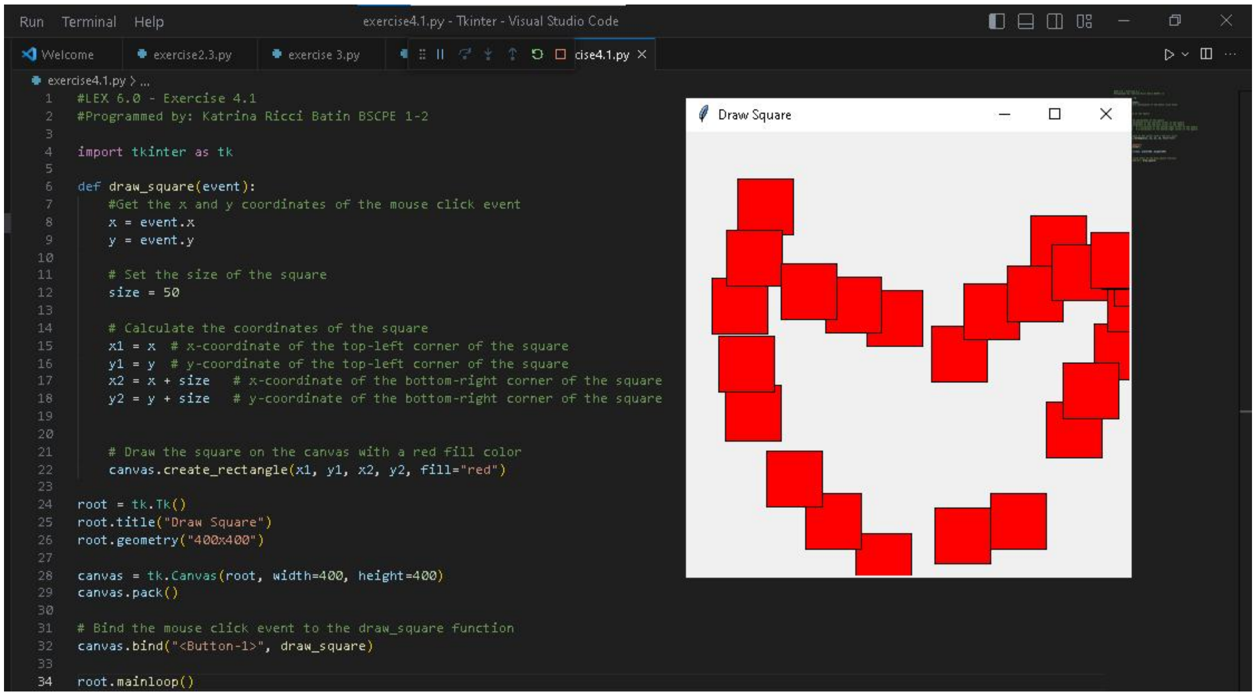
Exercise4.1: Draw a Square where the mouse is clicked Instead of always drawing the same shapes, use the mouse to draw a square: the top- left corner of the square goes where the mouse is clicked.

Code:



```
#LEX 6.0 - Exercise 4.1
#Programmed by: Katrina Ricci Batin BSCPE 1-2

import tkinter as tk

def draw_square(event):
    #Get the x and y coordinates of the mouse click event
    x = event.x
    y = event.y

    # Set the size of the square
    size = 50

    # Calculate the coordinates of the square
    x1 = x  # x-coordinate of the top-left corner of the square
    y1 = y  # y-coordinate of the top-left corner of the square
    x2 = x + size   # x-coordinate of the bottom-right corner of the square
    y2 = y + size   # y-coordinate of the bottom-right corner of the square


    # Draw the square on the canvas with a red fill color
    canvas.create_rectangle(x1, y1, x2, y2, fill="red")

root = tk.Tk()
root.title("Draw Square")
root.geometry("400x400")

canvas = tk.Canvas(root, width=400, height=400)
canvas.pack()

# Bind the mouse click event to the draw_square function
canvas.bind("<Button-1>", draw_square)

root.mainloop()
```

OUTPUT

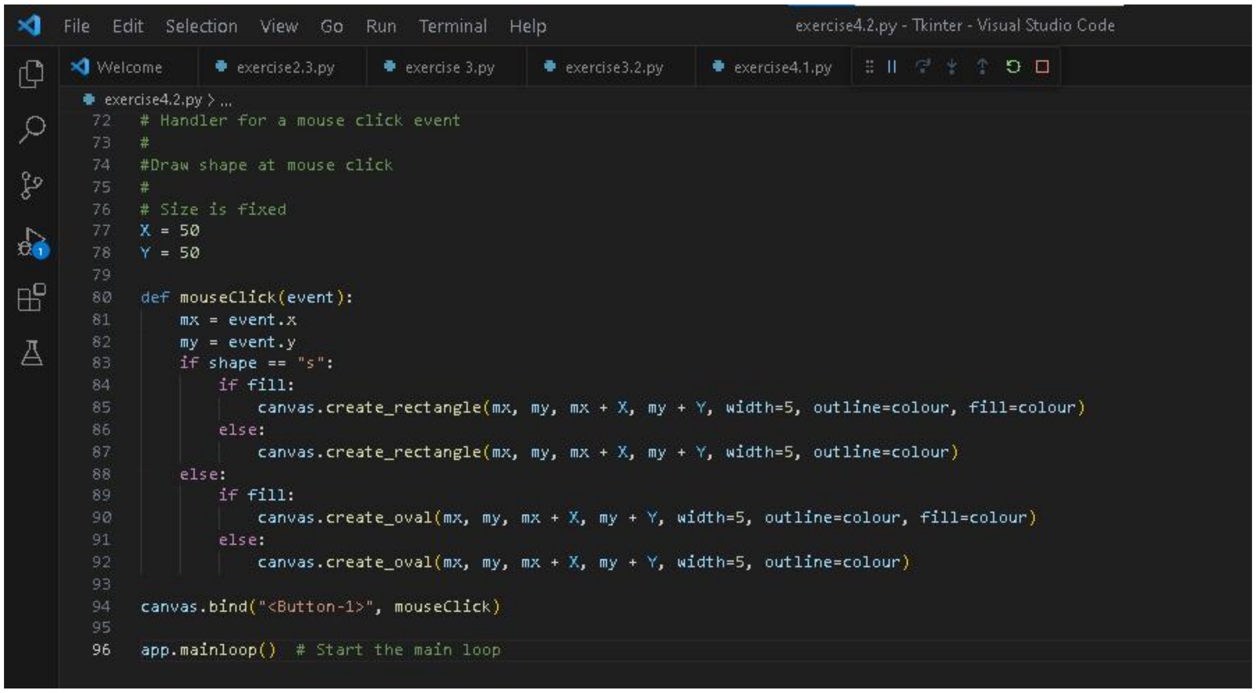


```python
#LEX 6.0 - Exercise 4.1
#Programmed by: Katrina Ricci Batin BSCPE 1-2

import tkinter as tk

def draw_square(event):
    #Get the x and y coordinates of the mouse click event
    x = event.x
    y = event.y

    # Set the size of the square
    size = 50

    # Calculate the coordinates of the square
    x1 = x  # x-coordinate of the top-left corner of the square
    y1 = y  # y-coordinate of the top-left corner of the square
    x2 = x + size   # x-coordinate of the bottom-right corner of the square
    y2 = y + size   # y-coordinate of the bottom-right corner of the square


    # Draw the square on the canvas with a red fill color
    canvas.create_rectangle(x1, y1, x2, y2, fill="red")

root = tk.Tk()
root.title("Draw Square")
root.geometry("400x400")

canvas = tk.Canvas(root, width=400, height=400)
canvas.pack()

# Bind the mouse click event to the draw_square function
canvas.bind("<Button-1>", draw_square)

root.mainloop()
```

Exercise 4.2: Change the shape, colour and fill.Use keys to specify the shape, colour and whether the shape is filled. For example:

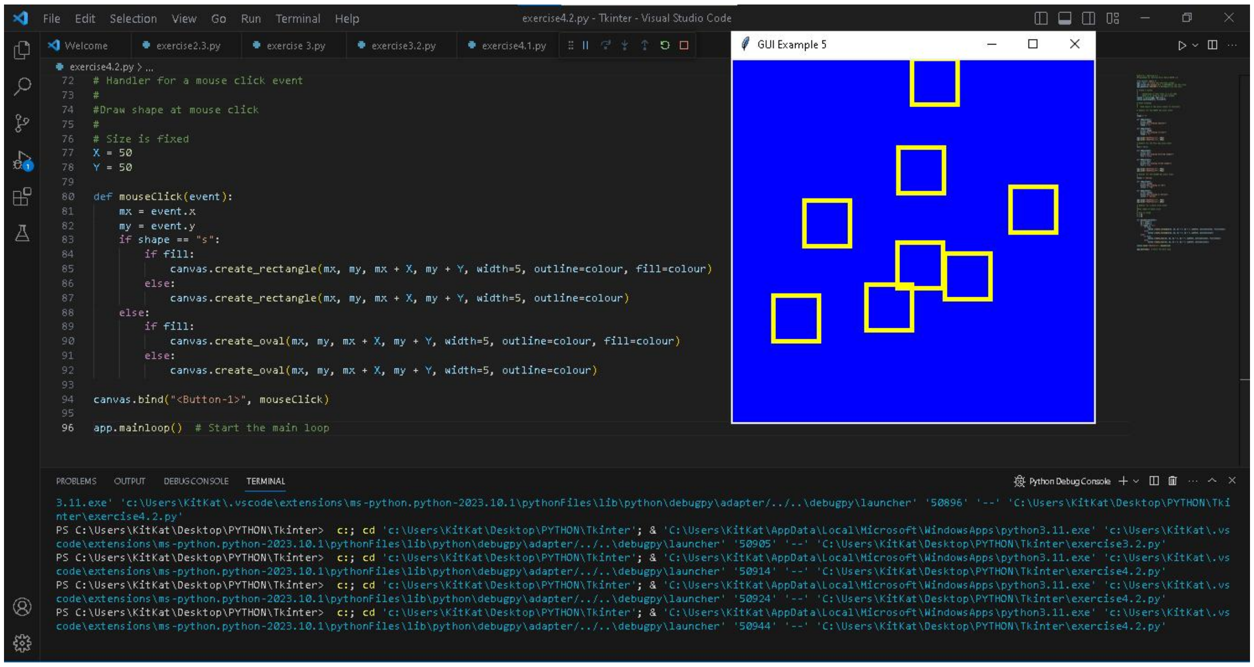
Shape: ‘s’ for square, ‘c’ for circle
Filling: ‘F’ for filled, ‘f’ for unfilled
Colour: ‘y’ for yellow, ‘r’ for red

Code:


File Edit Selection View Go Run Terminal Help
exercise4.2.py - Tkinter - Visual Studio Code
Welcome
exercise2.3.py
exercise 3.py
exercise3.2.py
exercise4.1.py
exercise4.2.py > ...

```python
# Handler for a mouse click event
#
#Draw shape at mouse click
#
# Size is fixed
X = 50
Y = 50

def mouseClick(event):
    mx = event.x
    my = event.y
    if shape == "s":
        if fill:
            canvas.create_rectangle(mx, my, mx + X, my + Y, width=5, outline=colour, fill=colour)
        else:
            canvas.create_rectangle(mx, my, mx + X, my + Y, width=5, outline=colour)
    else:
        if fill:
            canvas.create_oval(mx, my, mx + X, my + Y, width=5, outline=colour, fill=colour)
        else:
            canvas.create_oval(mx, my, mx + X, my + Y, width=5, outline=colour)

canvas.bind("<Button-1>", mouseClick)

app.mainloop()  # Start the main loop
```

OUTPUT



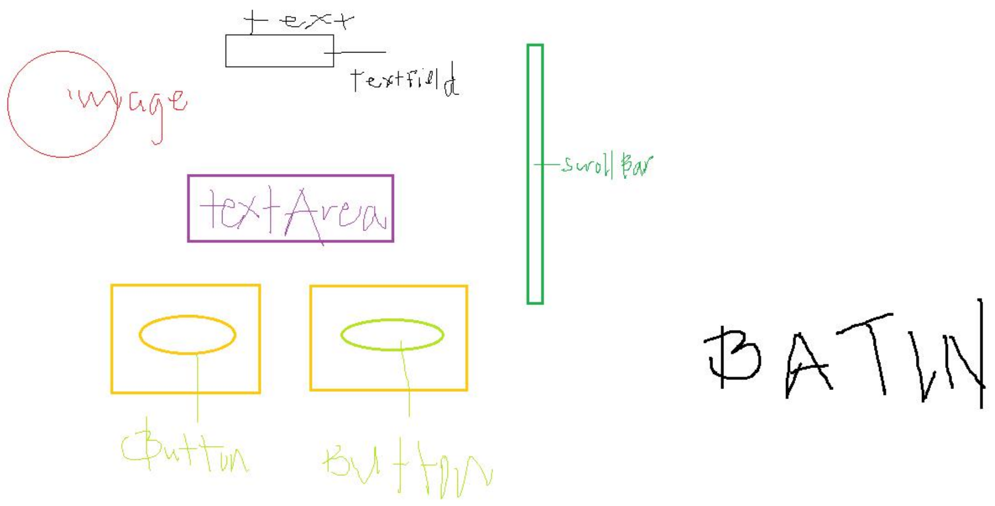
File Edit Selection View Go Run Terminal Help
exercise4.2.py - Tkinter - Visual Studio Code
Welcome
exercise2.3.py
exercise 3.py
exercise3.2.py
exercise4.1.py
GUI Example 5
exercise4.2.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python Debug Console

```python
# Handler for a mouse click event
#
#Draw shape at mouse click
#
# Size is fixed
X = 50
Y = 50

def mouseClick(event):
    mx = event.x
    my = event.y
    if shape == "s":
        if fill:
            canvas.create_rectangle(mx, my, mx + X, my + Y, width=5, outline=colour, fill=colour)
        else:
            canvas.create_rectangle(mx, my, mx + X, my + Y, width=5, outline=colour)
    else:
        if fill:
            canvas.create_oval(mx, my, mx + X, my + Y, width=5, outline=colour, fill=colour)
        else:
            canvas.create_oval(mx, my, mx + X, my + Y, width=5, outline=colour)

canvas.bind("<Button-1>", mouseClick)

app.mainloop()  # Start the main loop
```

```
3.11.exe' 'c:\Users\KitKat\.vscode\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '50896' '--' 'C:\Users\KitKat\Desktop\PYTHON\Tki
nter\exercise4.2.py'
PS C:\Users\KitKat\Desktop\PYTHON\Tkinter>  c:; cd 'c:\Users\KitKat\Desktop\PYTHON\Tkinter'; & 'C:\Users\KitKat\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\KitKat\.vs
code\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '50905' '--' 'C:\Users\KitKat\Desktop\PYTHON\Tkinter\exercise3.2.py'
PS C:\Users\KitKat\Desktop\PYTHON\Tkinter>  c:; cd 'c:\Users\KitKat\Desktop\PYTHON\Tkinter'; & 'C:\Users\KitKat\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\KitKat\.vs
code\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '50914' '--' 'C:\Users\KitKat\Desktop\PYTHON\Tkinter\exercise4.2.py'
PS C:\Users\KitKat\Desktop\PYTHON\Tkinter>  c:; cd 'c:\Users\KitKat\Desktop\PYTHON\Tkinter'; & 'C:\Users\KitKat\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\KitKat\.vs
code\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '50924' '--' 'C:\Users\KitKat\Desktop\PYTHON\Tkinter\exercise4.2.py'
PS C:\Users\KitKat\Desktop\PYTHON\Tkinter>  c:; cd 'c:\Users\KitKat\Desktop\PYTHON\Tkinter'; & 'C:\Users\KitKat\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\KitKat\.vs
code\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '50944' '--' 'C:\Users\KitKat\Desktop\PYTHON\Tkinter\exercise4.2.py'
```

Exercise 4.3: Interface Design Using a pencil and paper, sketch some better interfaces to draw shapes. Consider either a) how to show what the current drawing options are or b) alternative ways to specify the shape, colour and filling, plus other features that could be useful.

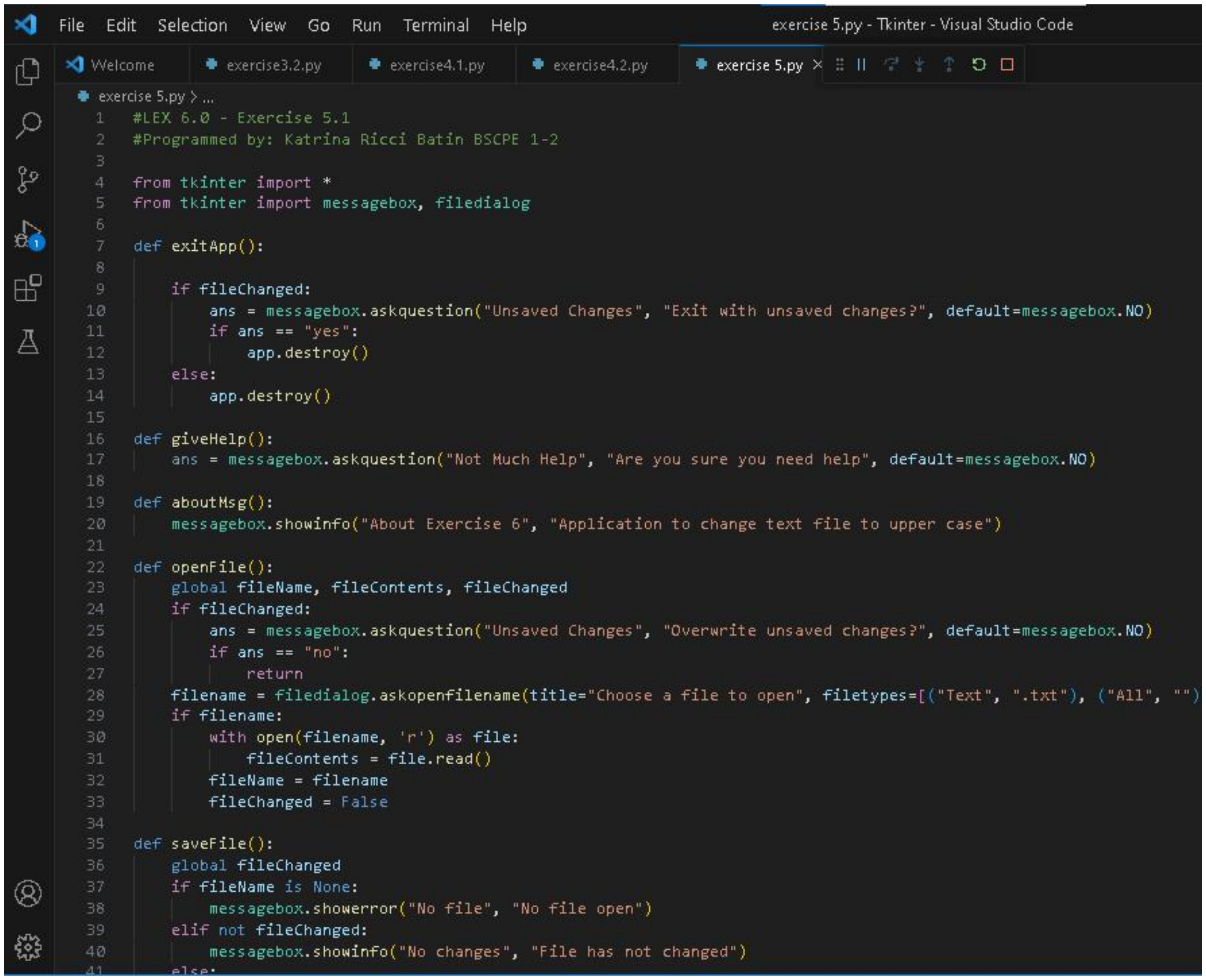
Output:



Exercise 5: Dialogs and Menu

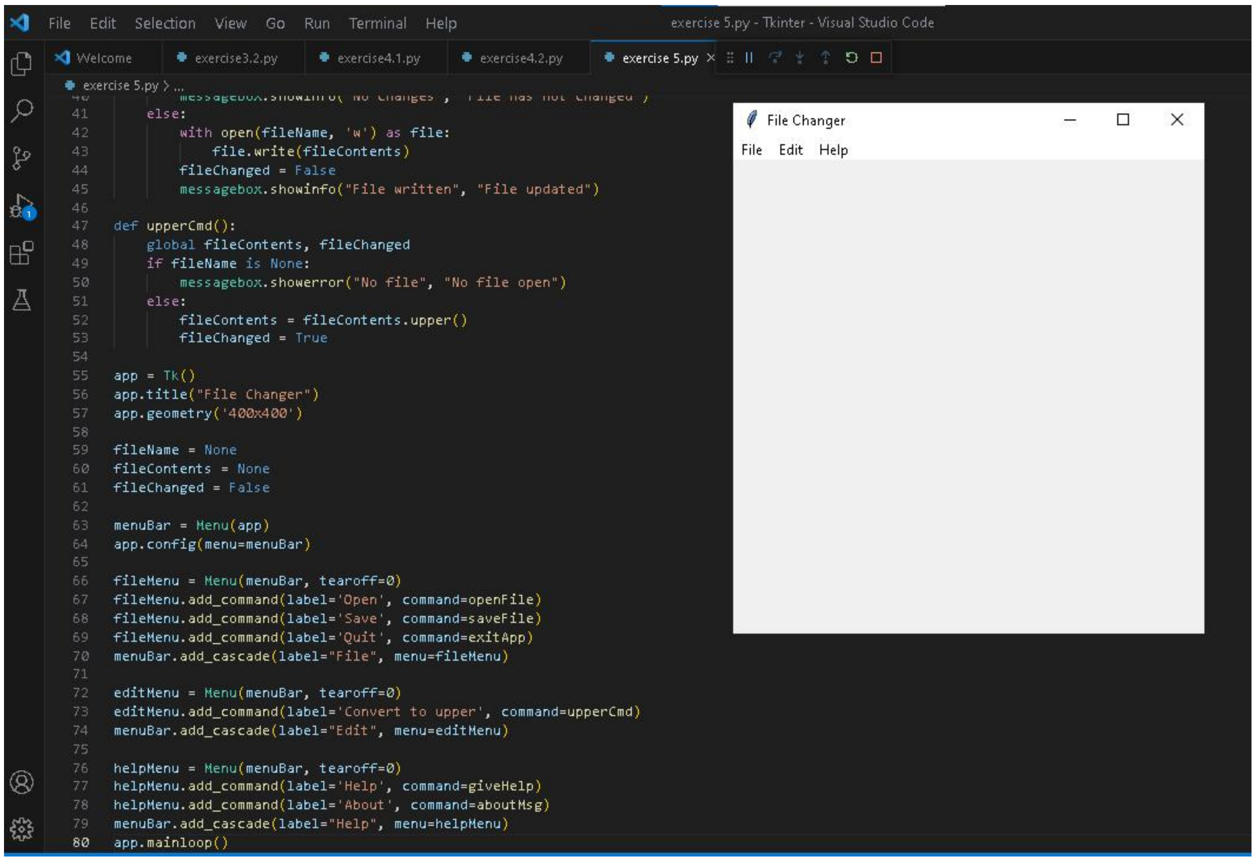
Exercise 5.1: Add menu items Add the new menu and menu items. At first, do not give a command.

Code:


File Edit Selection View Go Run Terminal Help
exercise 5.py - Tkinter - Visual Studio Code
Welcome
exercise3.2.py
exercise4.1.py
exercise4.2.py
exercise 5.py
exercise 5.py > ...

```python
#LEX 6.0 - Exercise 5.1
#Programmed by: Katrina Ricci Batin BSCPE 1-2

from tkinter import *
from tkinter import messagebox, filedialog

def exitApp():

    if fileChanged:
        ans = messagebox.askquestion("Unsaved Changes", "Exit with unsaved changes?", default=messagebox.NO)
        if ans == "yes":
            app.destroy()
    else:
        app.destroy()

def giveHelp():
    ans = messagebox.askquestion("Not Much Help", "Are you sure you need help", default=messagebox.NO)

def aboutMsg():
    messagebox.showinfo("About Exercise 6", "Application to change text file to upper case")

def openFile():
    global fileName, fileContents, fileChanged
    if fileChanged:
        ans = messagebox.askquestion("Unsaved Changes", "Overwrite unsaved changes?", default=messagebox.NO)
        if ans == "no":
            return
    filename = filedialog.askopenfilename(title="Choose a file to open", filetypes=[("Text", ".txt"), ("All", "")
    if filename:
        with open(filename, 'r') as file:
            fileContents = file.read()
        fileName = filename
        fileChanged = False

def saveFile():
    global fileChanged
    if fileName is None:
        messagebox.showerror("No file", "No file open")
    elif not fileChanged:
        messagebox.showinfo("No changes", "File has not changed")
    else:
```

OUTPUT



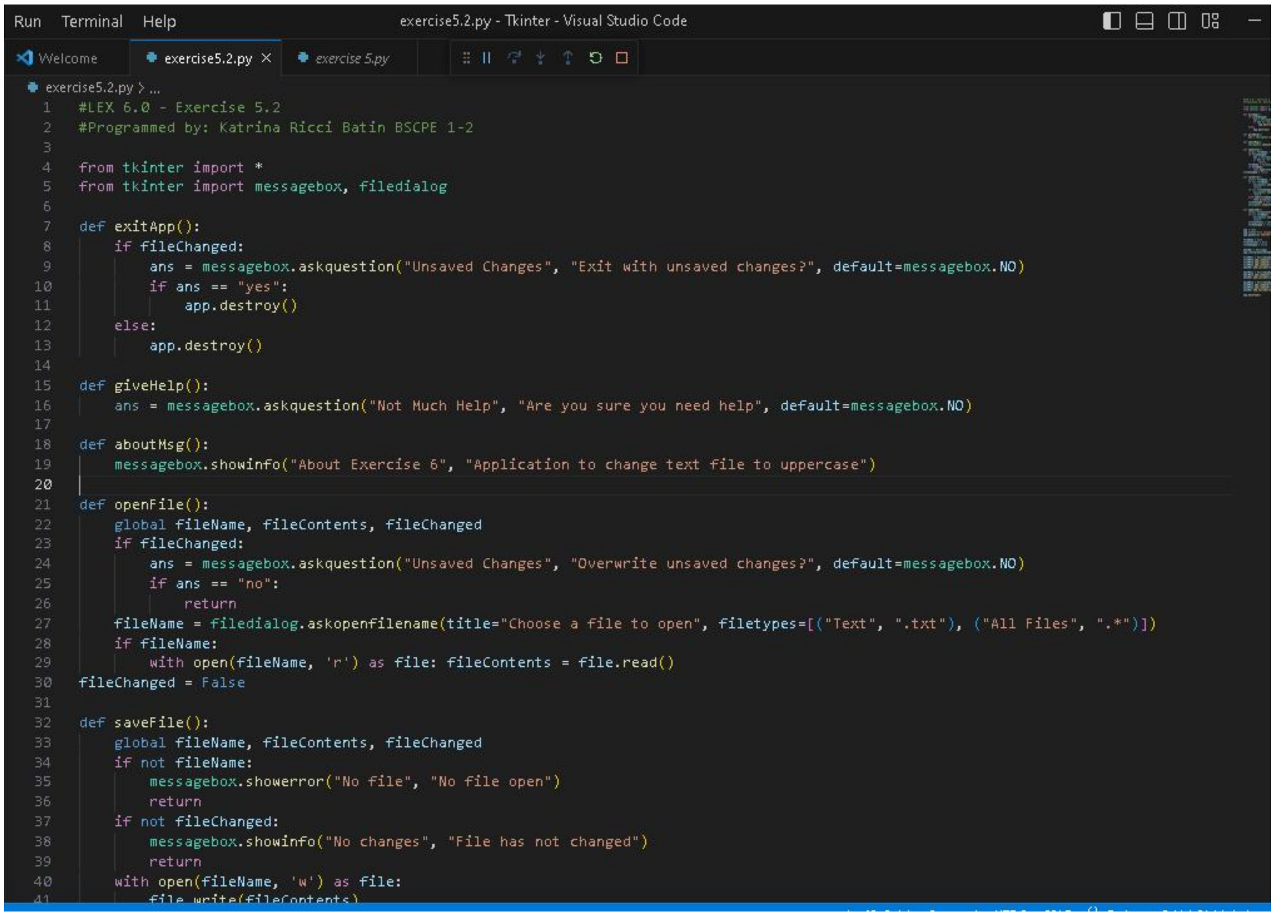
```
        messagebox.showinfo("No changes", "File has not changed")
    else:
        with open(fileName, 'w') as file:
            file.write(fileContents)
        fileChanged = False
        messagebox.showinfo("File written", "File updated")

def upperCmd():
    global fileContents, fileChanged
    if fileName is None:
        messagebox.showerror("No file", "No file open")
    else:
        fileContents = fileContents.upper()
        fileChanged = True

app = Tk()
app.title("File Changer")
app.geometry('400x400')

fileName = None
fileContents = None
fileChanged = False

menuBar = Menu(app)
app.config(menu=menuBar)

fileMenu = Menu(menuBar, tearoff=0)
fileMenu.add_command(label='Open', command=openFile)
fileMenu.add_command(label='Save', command=saveFile)
fileMenu.add_command(label='Quit', command=exitApp)
menuBar.add_cascade(label="File", menu=fileMenu)

editMenu = Menu(menuBar, tearoff=0)
editMenu.add_command(label='Convert to upper', command=upperCmd)
menuBar.add_cascade(label="Edit", menu=editMenu)

helpMenu = Menu(menuBar, tearoff=0)
helpMenu.add_command(label='Help', command=giveHelp)
helpMenu.add_command(label='About', command=aboutMsg)
menuBar.add_cascade(label="Help", menu=helpMenu)
app.mainloop()
```

Exercise 5.2: Implement Functions

Implement the functions to act on the command. You can use :

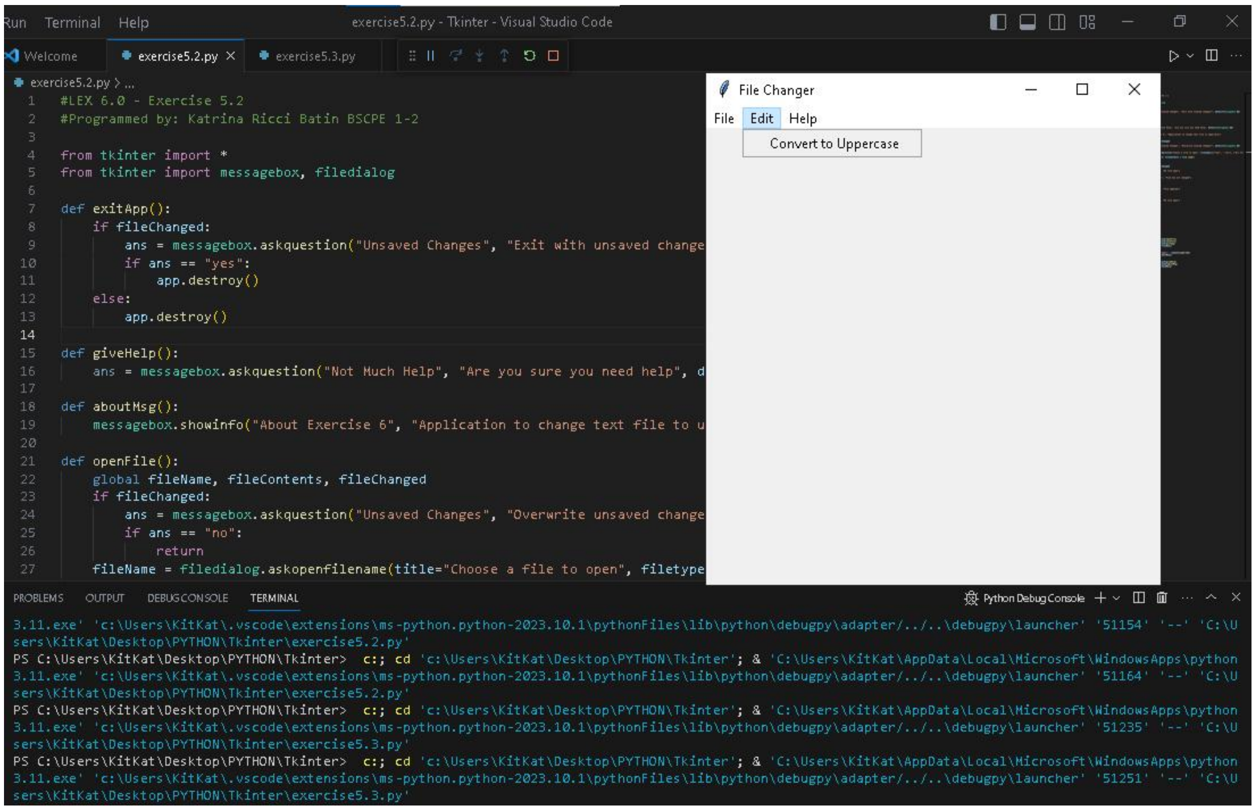
open(filename, mode) to open the file with mode 'r' and 'w'
f.close(0 to close a file
f.read() to read a whole file
s.upper() to convert a string s to uppercase (returns a new string)
f.write(string) to write a string to the file

Code:



```python
#LEX 6.0 - Exercise 5.2
#Programmed by: Katrina Ricci Batin BSCPE 1-2

from tkinter import *
from tkinter import messagebox, filedialog

def exitApp():
    if fileChanged:
        ans = messagebox.askquestion("Unsaved Changes", "Exit with unsaved changes?", default=messagebox.NO)
        if ans == "yes":
            app.destroy()
    else:
        app.destroy()

def giveHelp():
    ans = messagebox.askquestion("Not Much Help", "Are you sure you need help", default=messagebox.NO)

def aboutMsg():
    messagebox.showinfo("About Exercise 6", "Application to change text file to uppercase")

def openFile():
    global fileName, fileContents, fileChanged
    if fileChanged:
        ans = messagebox.askquestion("Unsaved Changes", "Overwrite unsaved changes?", default=messagebox.NO)
        if ans == "no":
            return
    fileName = filedialog.askopenfilename(title="Choose a file to open", filetypes=[("Text", ".txt"), ("All Files", ".*")])
    if fileName:
        with open(fileName, 'r') as file: fileContents = file.read()
fileChanged = False

def saveFile():
    global fileName, fileContents, fileChanged
    if not fileName:
        messagebox.showerror("No file", "No file open")
        return
    if not fileChanged:
        messagebox.showinfo("No changes", "File has not changed")
        return
    with open(fileName, 'w') as file:
        file.write(fileContents)
```

OUTPUT



```
#LEX 6.0 - Exercise 5.2
#Programmed by: Katrina Ricci Batin BSCPE 1-2

from tkinter import *
from tkinter import messagebox, filedialog

def exitApp():
    if fileChanged:
        ans = messagebox.askquestion("Unsaved Changes", "Exit with unsaved change
        if ans == "yes":
            app.destroy()
    else:
        app.destroy()

def giveHelp():
    ans = messagebox.askquestion("Not Much Help", "Are you sure you need help", d

def aboutMsg():
    messagebox.showinfo("About Exercise 6", "Application to change text file to u

def openFile():
    global fileName, fileContents, fileChanged
    if fileChanged:
        ans = messagebox.askquestion("Unsaved Changes", "Overwrite unsaved change
        if ans == "no":
            return
    fileName = filedialog.askopenfilename(title="Choose a file to open", filetype
```

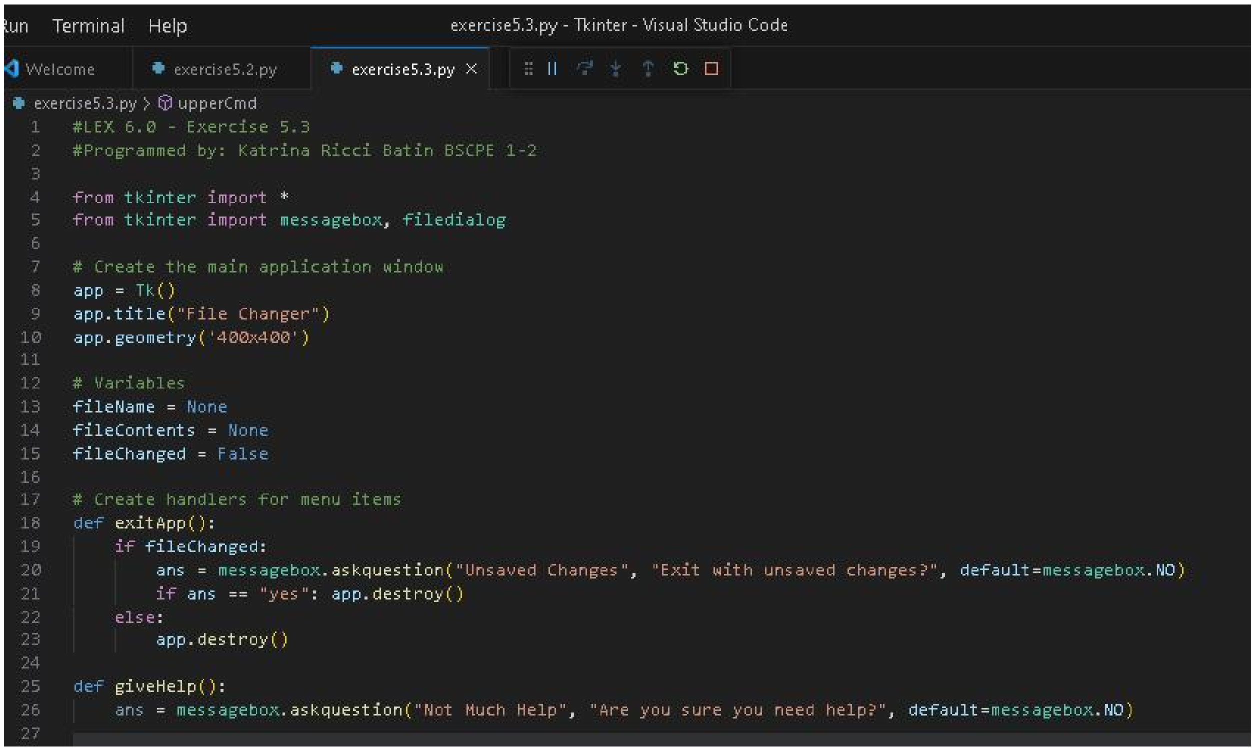
```
3.11.exe' 'c:\Users\KitKat\.vscode\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '51154' '--' 'C:\U
sers\KitKat\Desktop\PYTHON\Tkinter\exercise5.2.py'
PS C:\Users\KitKat\Desktop\PYTHON\Tkinter>  c:; cd 'c:\Users\KitKat\Desktop\PYTHON\Tkinter'; & 'C:\Users\KitKat\AppData\Local\Microsoft\WindowsApps\python
3.11.exe' 'c:\Users\KitKat\.vscode\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '51164' '--' 'C:\U
sers\KitKat\Desktop\PYTHON\Tkinter\exercise5.2.py'
PS C:\Users\KitKat\Desktop\PYTHON\Tkinter>  c:; cd 'c:\Users\KitKat\Desktop\PYTHON\Tkinter'; & 'C:\Users\KitKat\AppData\Local\Microsoft\WindowsApps\python
3.11.exe' 'c:\Users\KitKat\.vscode\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '51235' '--' 'C:\U
sers\KitKat\Desktop\PYTHON\Tkinter\exercise5.3.py'
PS C:\Users\KitKat\Desktop\PYTHON\Tkinter>  c:; cd 'c:\Users\KitKat\Desktop\PYTHON\Tkinter'; & 'C:\Users\KitKat\AppData\Local\Microsoft\WindowsApps\python
3.11.exe' 'c:\Users\KitKat\.vscode\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '51251' '--' 'C:\U
sers\KitKat\Desktop\PYTHON\Tkinter\exercise5.3.py'
```

Exercise 5.3: Add checks

Add checks so that a) the program never crashes and b) the user does not lose work. The followingtablesuggestwhichchecksareneeded.Displaysuitablemessagesineachcase.

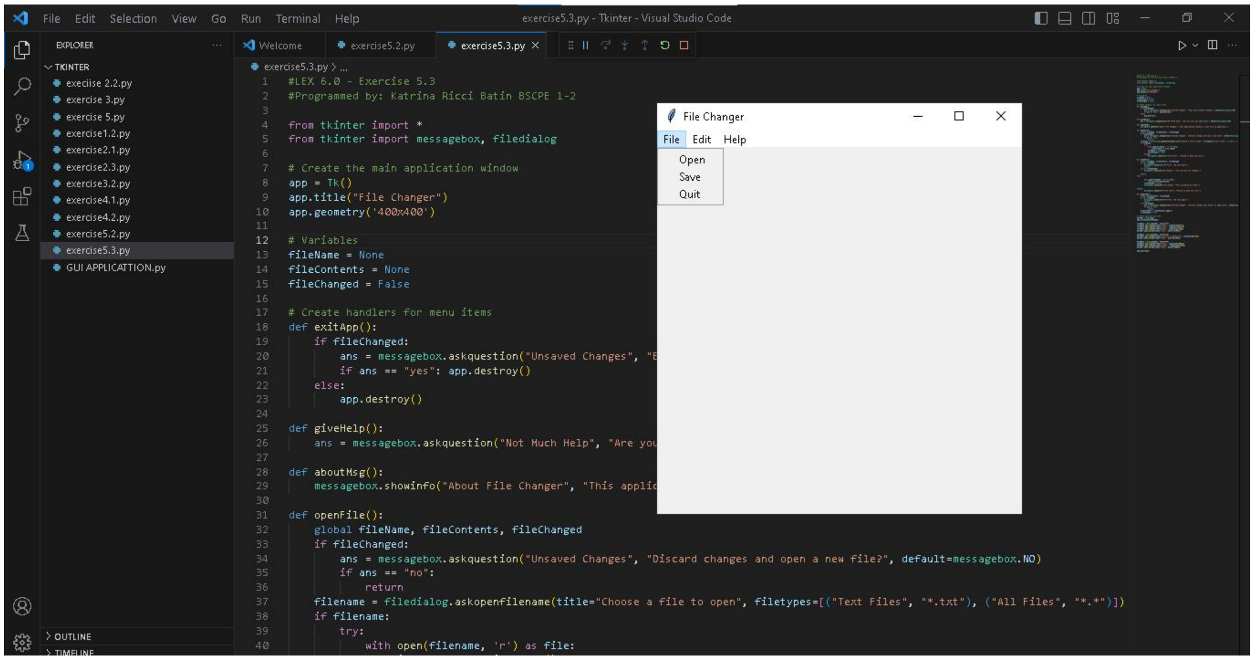
Command	**Checks Needed**
Open	Check for unsaved changes to the current file (Question)
Save	Checkafileisopen(Error)Checkthatchangesneedsaving(Info)
Quit	Check for unsaved changes to the current file (Question)
Convert to Upper	Check a file is open (Error)

CODE:



```python
#LEX 6.0 - Exercise 5.3
#Programmed by: Katrina Ricci Batin BSCPE 1-2

from tkinter import *
from tkinter import messagebox, filedialog

# Create the main application window
app = Tk()
app.title("File Changer")
app.geometry('400x400')

# Variables
fileName = None
fileContents = None
fileChanged = False

# Create handlers for menu items
def exitApp():
    if fileChanged:
        ans = messagebox.askquestion("Unsaved Changes", "Exit with unsaved changes?", default=messagebox.NO)
        if ans == "yes": app.destroy()
    else:
        app.destroy()

def giveHelp():
    ans = messagebox.askquestion("Not Much Help", "Are you sure you need help?", default=messagebox.NO)

```

OUTPUT



```python
#LEX 6.0 - Exercise 5.3
#Programmed by: Katrina Ricci Batin BSCPE 1-2

from tkinter import *
from tkinter import messagebox, filedialog

# Create the main application window
app = Tk()
app.title("File Changer")
app.geometry('400x400')

# Variables
fileName = None
fileContents = None
fileChanged = False

# Create handlers for menu items
def exitApp():
    if fileChanged:
        ans = messagebox.askquestion("Unsaved Changes", "E
        if ans == "yes": app.destroy()
    else:
        app.destroy()

def giveHelp():
    ans = messagebox.askquestion("Not Much Help", "Are you

def aboutMsg():
    messagebox.showinfo("About File Changer", "This applic

def openFile():
    global fileName, fileContents, fileChanged
    if fileChanged:
        ans = messagebox.askquestion("Unsaved Changes", "Discard changes and open a new file?", default=messagebox.NO)
        if ans == "no":
            return
    filename = filedialog.askopenfilename(title="Choose a file to open", filetypes=[("Text Files", "*.txt"), ("All Files", "*.*")])
    if filename:
        try:
            with open(filename, 'r') as file:
```