

Course Name	ITD 2313 – Script Programming
Instructor	Andy Tripp
Student Name	Timothy Obinda
Due date	11/09/2025
Grade	Put grade earned here
Grading Comments	Put instructor comments here

INSTRUCTIONS FOR THE EXERCISE

You should always read the instructions in full. It is best to do a full read through before starting the assignment. Each screenshot needs to be appropriately labeled.

In these instructions, you are given information to execute specific examples in the text. You are given the Section, subsection, and a page number to identify a set of steps. On each page listed, there will be 1 or more numbered tasks to perform. These numbered tasks will be what you are to type in, execute, and then grab the screen shots of. Those screenshots will then go into your submission document.

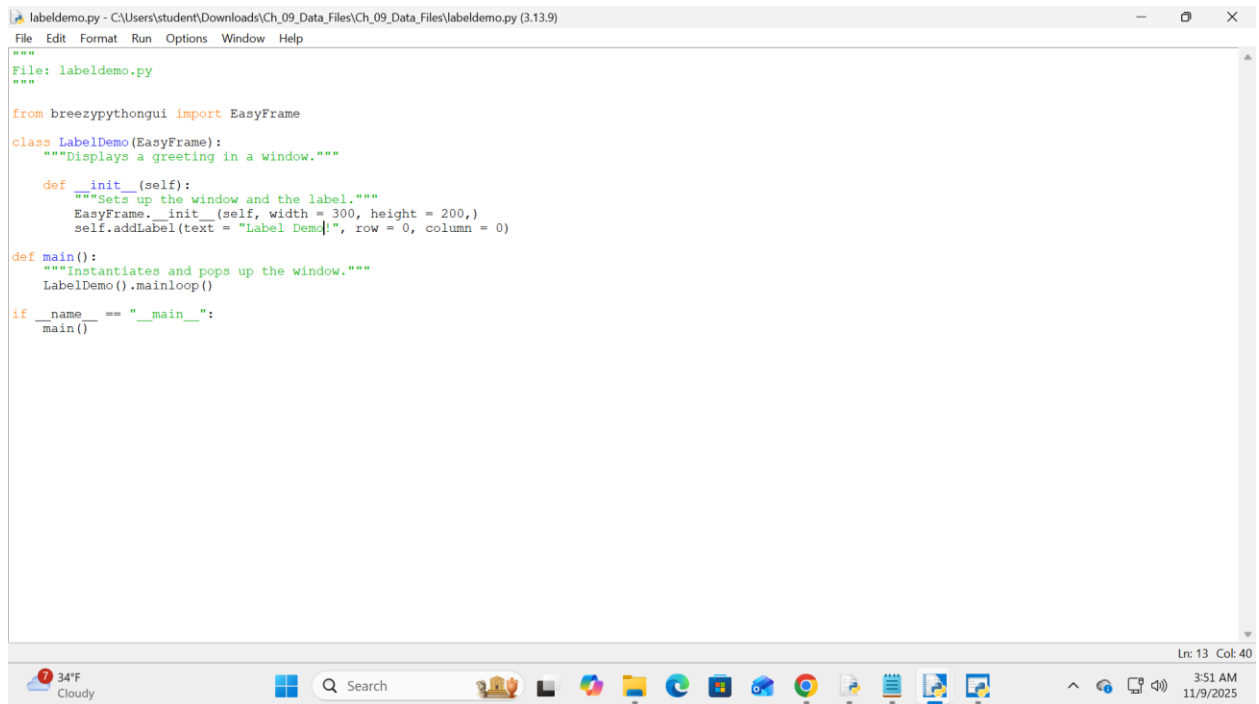
Some advice, copy this instruction set into your submission document and then put the screen shots under each numbered task. Each individual book page in the instructions should be in a different screen shot. For any single book page, you may have all the numbered tasks on that page to be in a single screen shot.

Windows and Window Components

Windows Layout

Page 230-231

1. Use the code example block on the bottom of page 230 to start this set of instructions. You may need to review the previous pages to set up the GUI environment in Python correctly. Show in this first screenshot the grid layout results. Expand out the window to show the stuck in the corner the text mentions.



```
labeldemo.py - C:\Users\student\Downloads\Ch_09_Data_Files\Ch_09_Data_Files\labeldemo.py (3.13.9)
File Edit Format Run Options Window Help
"""
File: labeldemo.py
"""

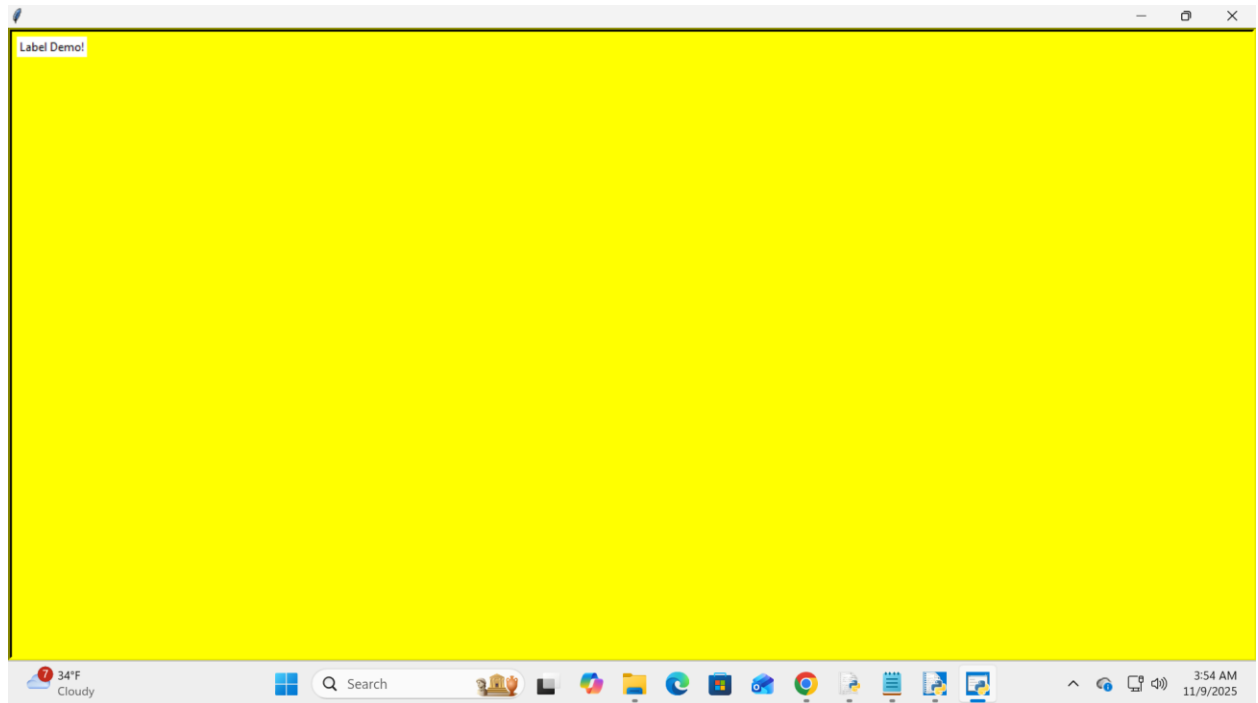
from breezypythongui import EasyFrame

class LabelDemo(EasyFrame):
    """Displays a greeting in a window."""

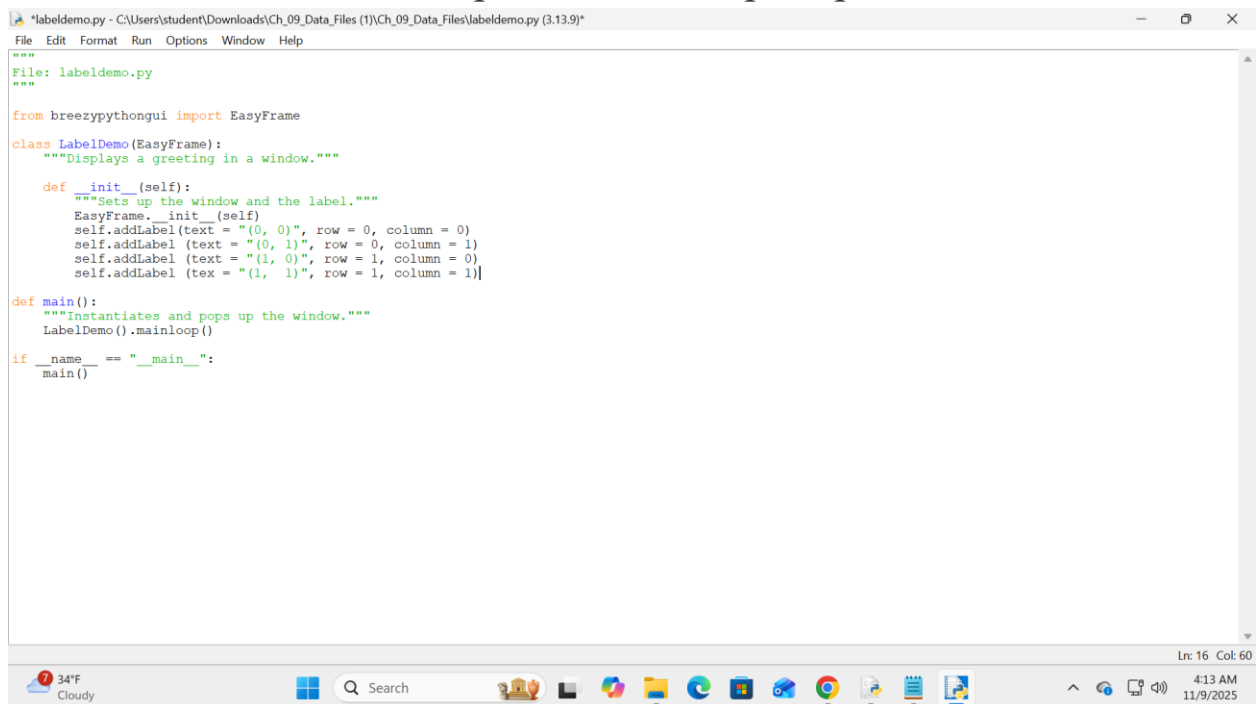
    def __init__(self):
        """Sets up the window and the label."""
        EasyFrame.__init__(self, width = 300, height = 200,)
        self.addLabel(text = "Label Demo!", row = 0, column = 0)

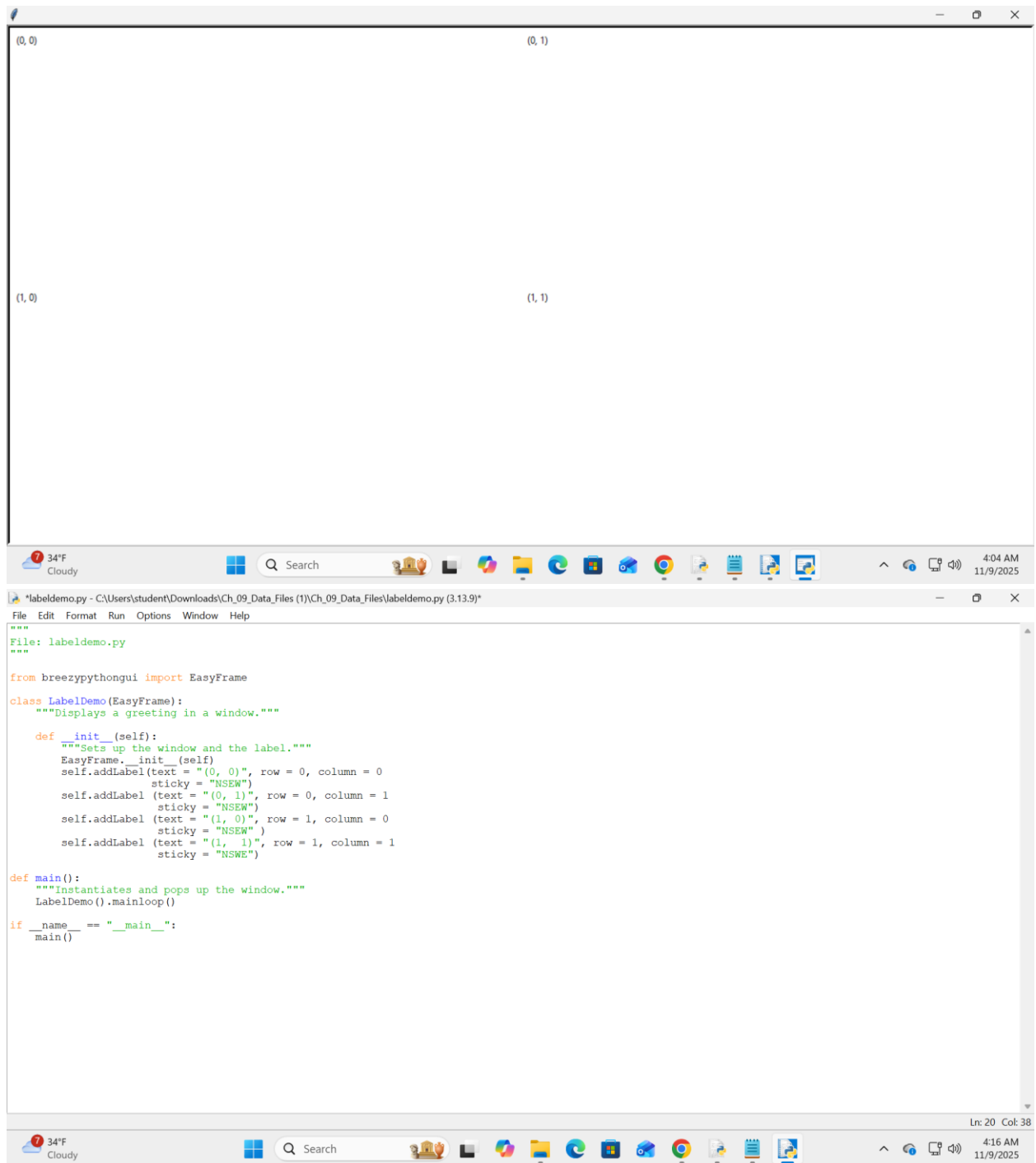
def main():
    """Instantiates and pops up the window."""
    LabelDemo().mainloop()

if __name__ == "__main__":
    main()
```



2. Using the code example blocks in the middle and at the bottom of page 231, change the look and feel to that of the centered look that is shown on the top of page 232. Expand out the window to show the difference between step 1 and this step output.





```
labeldemo.py - C:\Users\student\Downloads\Ch_09_Data_Files(1)\Ch_09_Data_Files\labeldemo.py (3.13.9)
File Edit Format Run Options Window Help
"""
File: labeldemo.py
"""

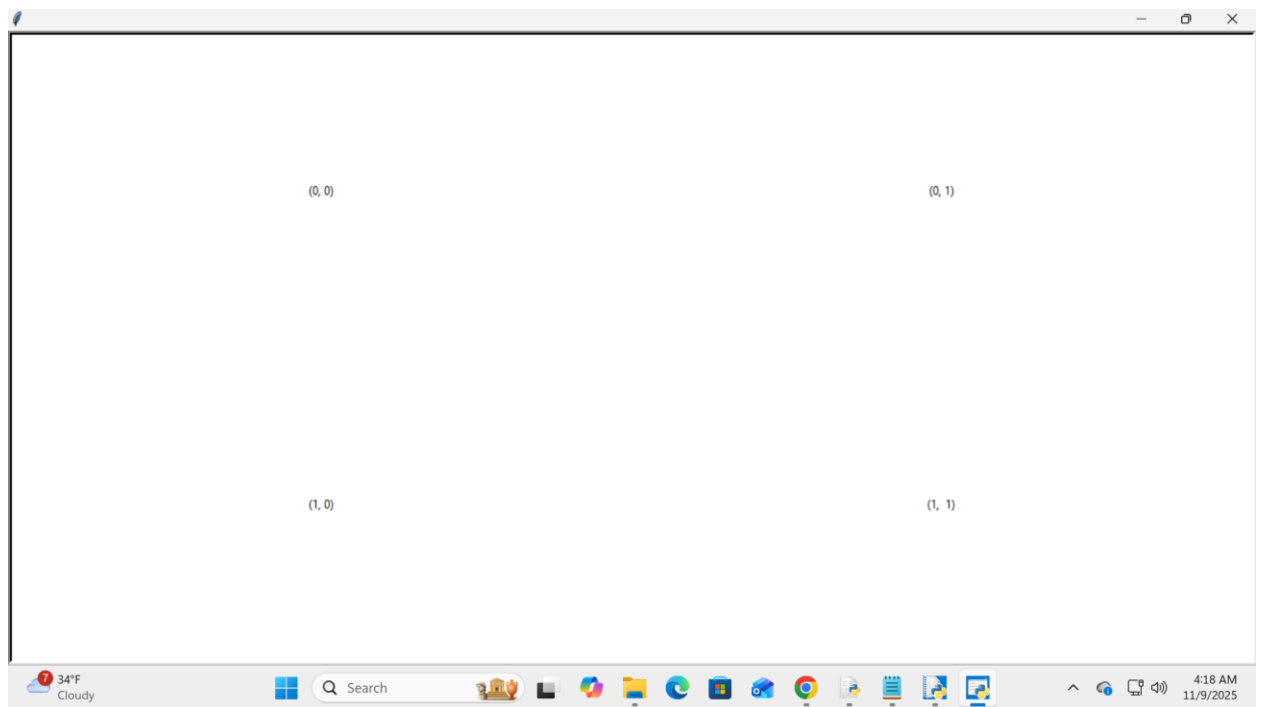
from breezypythongui import EasyFrame

class LabelDemo(EasyFrame):
    """Displays a greeting in a window."""

    def __init__(self):
        """Sets up the window and the label."""
        EasyFrame.__init__(self)
        self.addLabel(text = "(0, 0)", row = 0, column = 0,
            sticky = "NSEW")
        self.addLabel (text = "(0, 1)", row = 0, column = 1,
            sticky = "NSEW")
        self.addLabel (text = "(1, 0)", row = 1, column = 0,
            sticky = "NSEW")
        self.addLabel (text = "(1, 1)", row = 1, column = 1,
            sticky = "NSEW")

    def main():
        """Instantiates and pops up the window."""
        LabelDemo().mainloop()

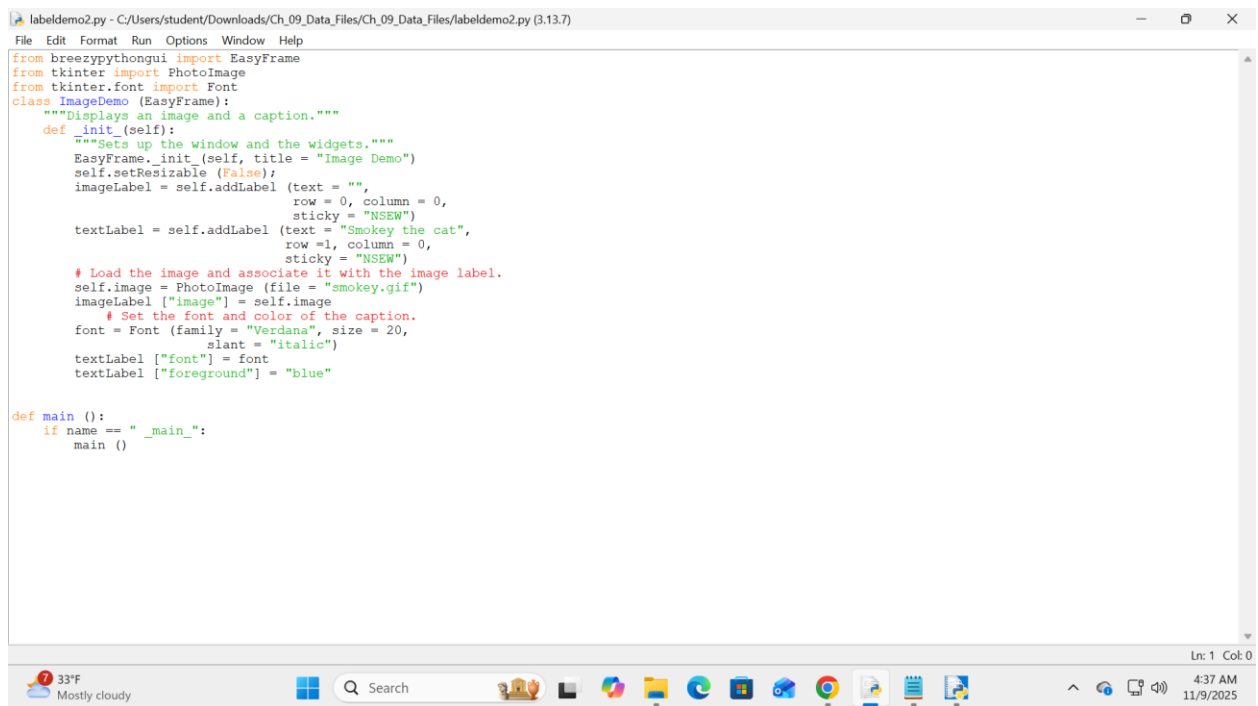
if __name__ == "__main__":
    main()
```



Displaying Images

Page 233 – 234

1. Smokey the Cat is back. Using the code example block on 233 and spills over to the top of page 234, get the Smokey picture to show up like it does at the top of page 233.



```
labeldemo2.py - C:/Users/student/Downloads/Ch_09_Data_Files/Ch_09_Data_Files/labeldemo2.py (3.13.7)
File Edit Format Run Options Window Help
from breezypythongui import EasyFrame
from tkinter import PhotoImage
from tkinter.font import Font
class ImageDemo (EasyFrame):
    """Displays an image and a caption."""
    def __init__(self):
        """Sets up the window and the widgets."""
        EasyFrame.__init__(self, title = "Image Demo")
        self.setResizable (False)
        imageLabel = self.addLabel (text = "",
                                    row = 0, column = 0,
                                    sticky = "NSEW")

        textLabel = self.addLabel (text = "Smokey the cat",
                                   row = 1, column = 0,
                                   sticky = "NSEW")

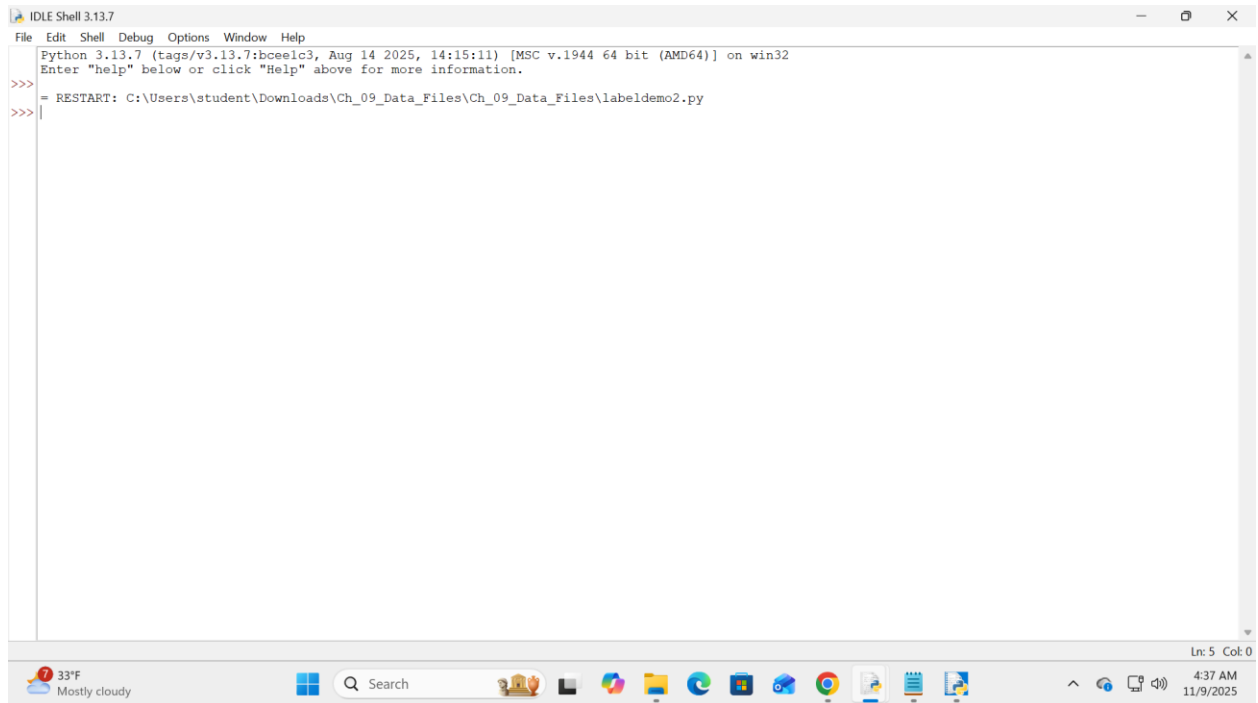
        # Load the image and associate it with the image label.
        self.image = PhotoImage (file = "smokey.gif")
        imageLabel ["image"] = self.image

        # Set the font and color of the caption.
        font = Font (family = "Verdana", size = 20,
                    slant = "italic")
        textLabel ["font"] = font
        textLabel ["foreground"] = "blue"

def main ():
    if name == "_main_":
        main ()
```

Ln: 1 Col: 0

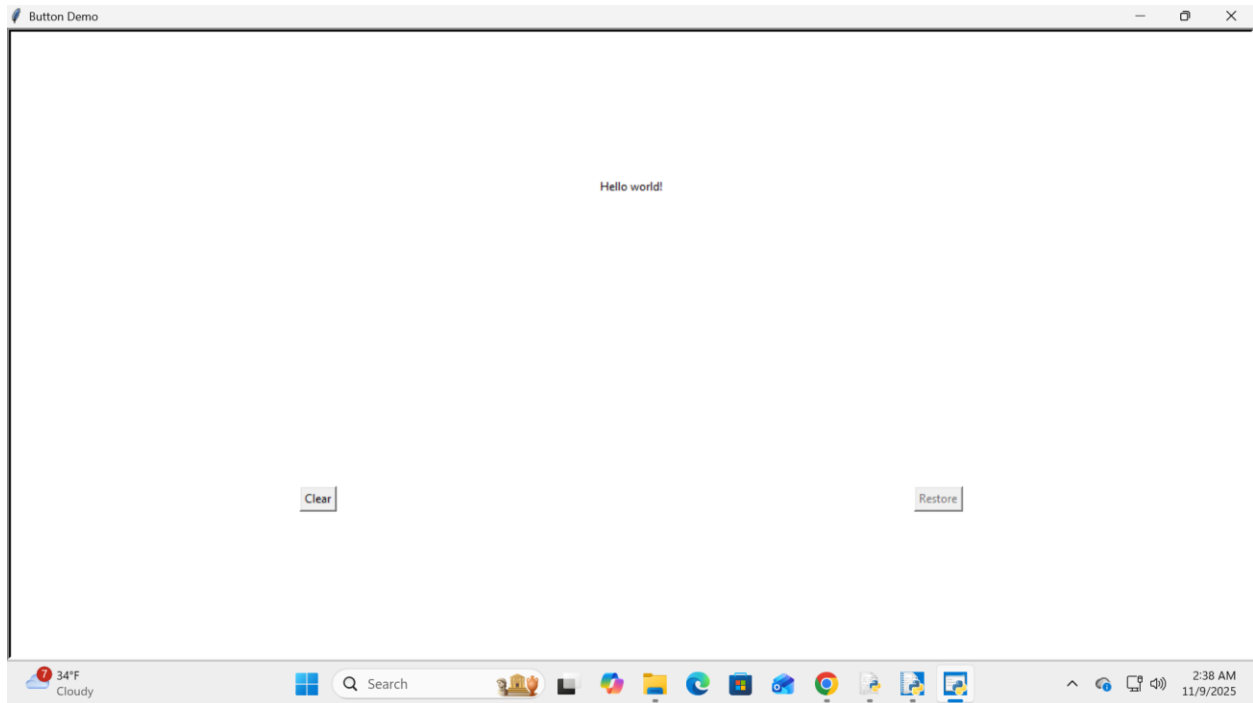
33°F Mostly cloudy 4:37 AM 11/9/2025



Command Buttons and Responding to events

Pages 235 – 236

1. Use the code example blocks on the pages to get the Hello World program and output seen on page 235 in the middle of the page.



Input and Output with Entry Fields

Text Fields

Page 237 – 238

1. Using the code example blocks on these pages, produce the input/output example on the middle page 237. Grab the screenshot with a phrase that is your favorite inspirational quote of all time instead of the phrase seen in the text.


```
textfielddemo.py - C:\Users\student\Downloads\Ch_09_Data_Files\Ch_09_Data_Files\textfielddemo.py (3.13.9)
File Edit Format Run Options Window Help
"""
File: textfielddemo.py
"""

from breezypythongui import EasyFrame

class TextFieldDemo(EasyFrame):
    """Converts an input string to uppercase and displays the result."""

    def __init__(self):
        """Sets up the window and widgets."""
        EasyFrame.__init__(self, title = "Text Field Demo")

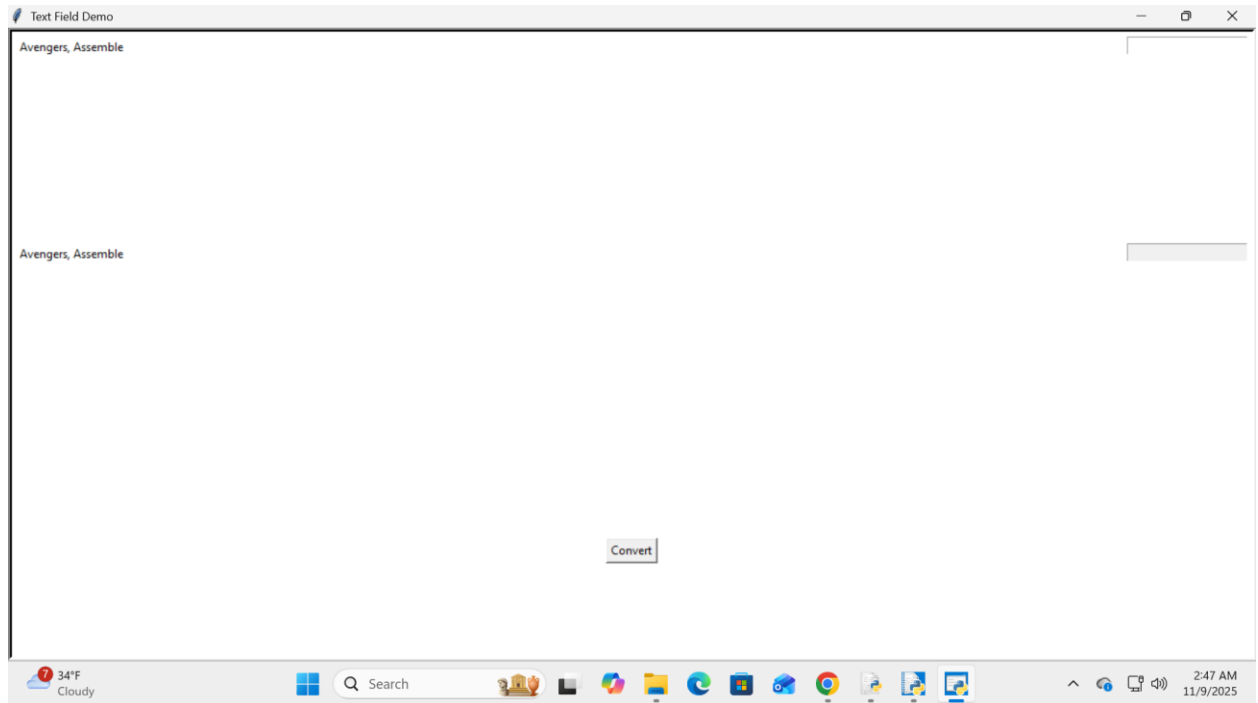
        # Label and field for the input
        self.addLabel(text = "Avengers, Assemble",
                      row = 0, column = 0)
        self.inputField = self.addTextField(text = "",
                                           row = 0,
                                           column = 1)

        # Label and field for the output
        self.addLabel(text = "Avengers, Assemble",
                      row = 1, column = 0)
        self.outputField = self.addTextField(text = "",
                                           row = 1,
                                           column = 1,
                                           state = "readonly")

        # The command button
        self.button = self.addButton(text = "Convert",
                                     row = 2, column = 0,
                                     colspan = 2,
                                     command = self.convert)

        # The event handling method for the button
    def convert(self):
        """Inputs the string, converts it to uppercase,
        and outputs the result."""
        text = self.inputField.getText()
        result = text.upper()
        self.outputField.setText(result)

Ln: 18 Col: 0
```



Integer and Float Fields for Numeric Data

Pages 238 – 239

1. Using the code example blocks on these two pages, produce the output seen on the bottom of page 238 except use the number 1941.002 as the number entered and show in the program instead of what is shown in the text.

```
numberfielddemo.py - C:\Users\student\Downloads\Ch_09_Data_Files\Ch_09_Data_Files\numberfielddemo.py (3.13.9)
File Edit Format Run Options Window Help

def __init__(self):
    """sets up the window and widgets."""
    EasyFrame.__init__(self, title = "Number Field Demo")

    # Label and field for the input
    self.addLabel(text = "An integer",
                  row = 0, column = 0)
    self.inputField = self.addIntegerField(value = 1941.002,
                                           row = 0,
                                           column = 1,
                                           width = 10)

    # Label and field for the output
    self.addLabel(text = "Square root",
                  row = 1, column = 0)
    self.outputField = self.addFloatField(value = 0.0,
                                           row = 1,
                                           column = 1,
                                           width = 8,
                                           precision = 2,
                                           state = "readonly")

    # The command button
    self.addButton(text = "Compute", row = 2, column = 0,
                   colspan = 2, command = self.computeSqrt)

    # The event handling method for the button
    def computeSqrt(self):
        """inputs the integer, computes the square root,
        and outputs the result."""
        number = self.inputField.getNumber()
        result = math.sqrt(number)
        self.outputField.setNumber(result)

def main():
    """Instantiate and pop up the window."""
    NumberFieldDemo().mainloop()

if __name__ == "__main__":
    main()
```

Ln: 18 Col: 63

34°F Cloudy 2:58 AM 11/9/2025

```
numberfielddemo.py - C:\Users\student\Downloads\Ch_09_Data_Files\Ch_09_Data_Files\numberfielddemo.py (3.13.9)
File Edit Shell Debug Options Window Help

Pyt AMI Ent
>>> An integer 1941.002
     Square root 0.0
     Compute

Click "Help" above for more information.
student\Downloads\Ch_09_Data_Files\Ch_09_Data_Files\numberfi

def
if
```

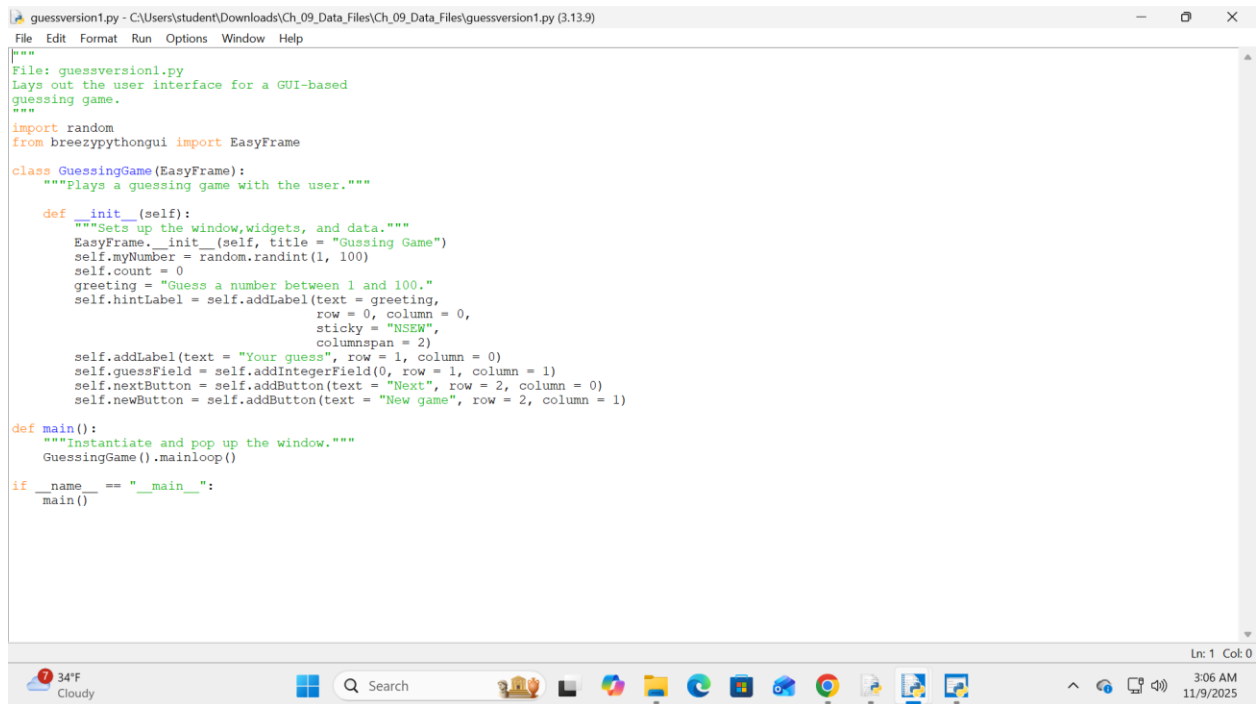
Ln: 18 Col: 63

34°F Cloudy 2:58 AM 11/9/2025

Fail-Safe Programming

Page 245 – 246

1. Using the code example blocks on the pages, produce the output shown on page 246. You may use the text book's example for this one if you so choose.



```
guessversion1.py - C:\Users\student\Downloads\Ch_09_Data_Files\Ch_09_Data_Files\guessversion1.py (3.13.9)
File Edit Format Run Options Window Help
"""
File: guessversion1.py
Lays out the user interface for a GUI-based
guessing game.
"""
import random
from breezypythongui import EasyFrame

class GuessingGame(EasyFrame):
    """Plays a guessing game with the user."""

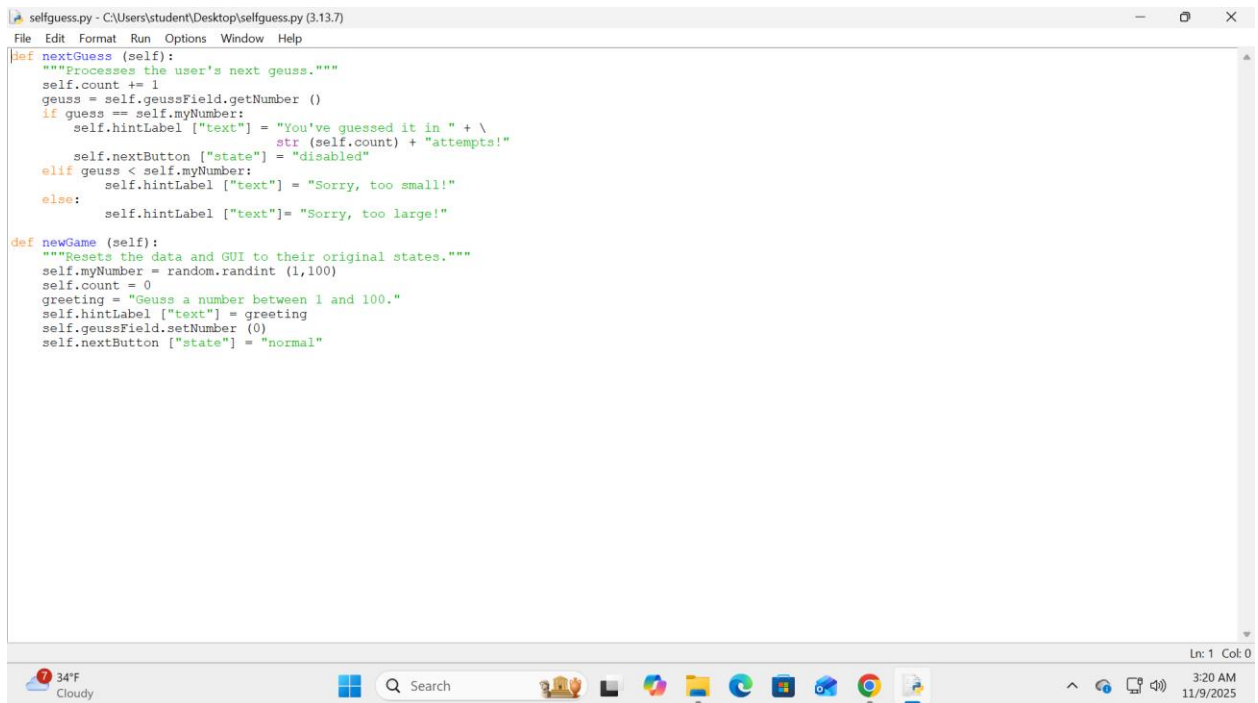
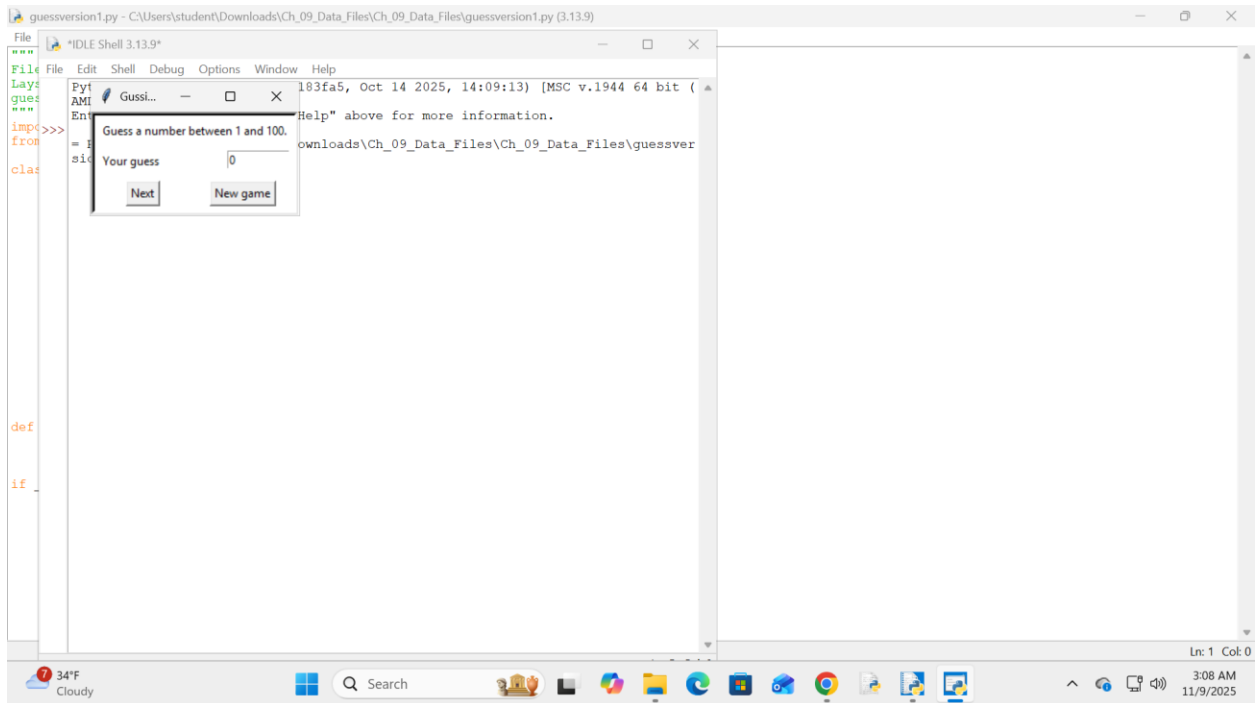
    def __init__(self):
        """Sets up the window, widgets, and data."""
        EasyFrame.__init__(self, title = "Guessing Game")
        self.myNumber = random.randint(1, 100)
        self.count = 0
        greeting = "Guess a number between 1 and 100."
        self.hintLabel = self.addLabel(text = greeting,
                                       row = 0, column = 0,
                                       sticky = "NSEW",
                                       colspan = 2)
        self.addLabel(text = "Your guess", row = 1, column = 0)
        self.guessField = self.addIntegerField(0, row = 1, column = 1)
        self.nextButton = self.addButton(text = "Next", row = 2, column = 0)
        self.newButton = self.addButton(text = "New game", row = 2, column = 1)

def main():
    """Instantiate and pop up the window."""
    GuessingGame().mainloop()

if __name__ == "__main__":
    main()
```

Ln: 1 Col: 0

34°F Cloudy 3:06 AM 11/9/2025



```
IDLE Shell 3.13.7
File Edit Shell Debug Options Window Help
Python 3.13.7 (tags/v3.13.7:bceec3, Aug 14 2025, 14:15:11) [MSC v.1944 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>>
>>> ===== RESTART: C:\Users\student\Desktop\selfguess.py =====
>>> |
```

Ln: 5 Col: 0

34°F Cloudy 3:21 AM 11/9/2025

You may find the other examples of code in the chapter to be very useful to work through but no screen shots will be required of them. You are highly encouraged to experiment with the various topics that are not specifically assigned in this assignment.