

การทดลองที่ CE-OOP 06

เรื่อง Graphic User Interface: GUI

วัตถุประสงค์

1. เพื่อให้ผู้อ่านได้เรียนรู้และเข้าใจเจตนาการสร้าง GUI
2. เพื่อให้ผู้อ่านได้เรียนรู้และฝึกฝนการเขียนโปรแกรมสร้าง GUI ด้วยภาษา Python
3. เพื่อให้ผู้อ่านนำความรู้เกี่ยวกับการสร้าง Class มาใช้งานร่วมกับการสร้าง GUI

เอกสารฉบับนี้แบ่งออกเป็น 3 ส่วนที่สำคัญนั่นคือ ทฤษฎีที่เกี่ยวข้อง พื้นฐานการสร้าง GUI การประยุกต์ใช้ GUI กับสถิติ

ตอนที่ 1 ทฤษฎีที่เกี่ยวข้อง

การสร้างส่วนประสานกับผู้ใช้แบบกราฟิก (Graphic User Interface: GUI) คือการทำให้ผู้ใช้งานทำงานได้ง่ายและสะดวกสบายด้วยกราฟิก สาเหตุที่ต้องสร้าง GUI เป็นเพราะการทำงานกับโค้ดโดยตรงนั้น ผู้ใช้งานต้องมีความเข้าใจและความชำนาญการใช้งานโปรแกรมในขั้นที่ดี แต่ผู้ใช้งานทั่วไปไม่อาจทำงานกับโปรแกรมนี้ได้ การสร้างเป็น GUI จะส่งผลให้ผู้ใช้งานทุกคนทำงานสามารถกับโปรแกรมนั้นได้อย่างง่ายดาย ในรูปที่ 1-1 คือการเปิดดูข้อมูลในไดรฟ์ D ด้วยการพิมพ์คำสั่ง (Command line) คำสั่งนั้นคือ D: dir

```

Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Naphasrapee>D:

D:\>dir
Volume in drive D is Data
Volume Serial Number is 1480-8B5C

Directory of D:\

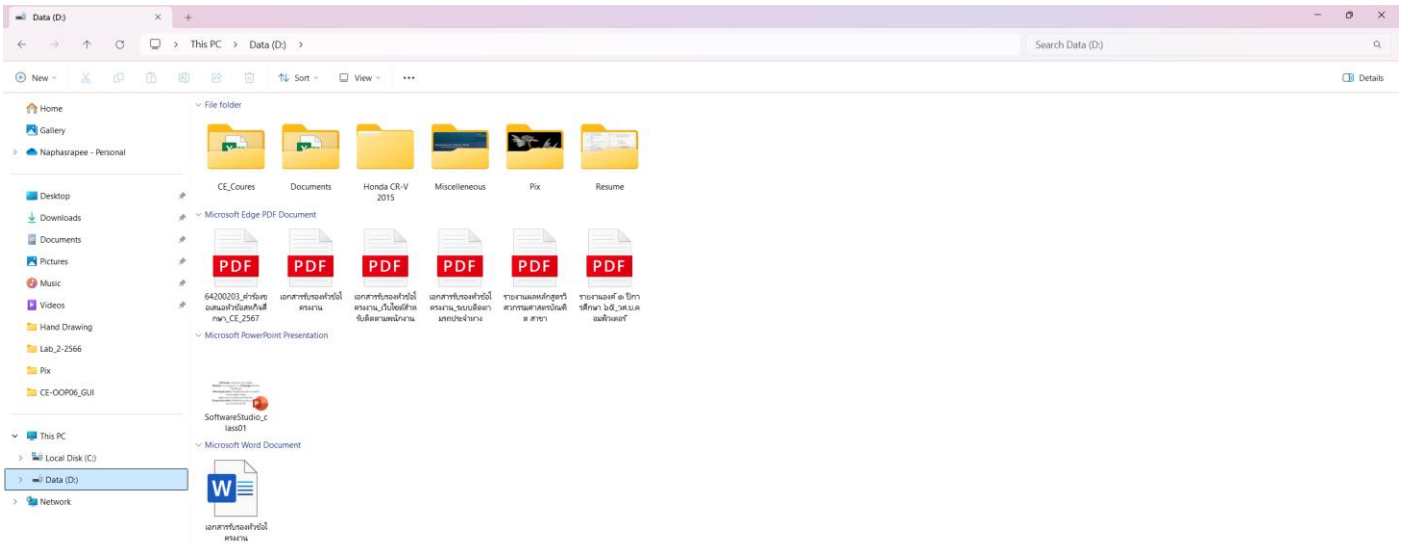
08/03/2024  09:38 PM           149,162  64200203_คำร้องขอเสนอ...วช อสหก จสึ ษา_CE_2567.pdf
02/05/2025  03:27 PM          <DIR>          CE_Coures
02/18/2025  12:44 PM          <DIR>          Documents
02/17/2025  10:47 PM          <DIR>          Honda CR-V 2015
02/05/2025  04:03 PM          <DIR>          Miscellaneous
03/01/2025  04:29 PM          <DIR>          Pix
11/22/2024  04:40 PM          <DIR>          Resume
12/02/2024  11:22 AM           50,543  SoftwareStudio_class01.pptx
08/30/2024  11:07 AM          443,735  รายงานผลหลั กสู ตรวิ ศวกรรมศาสตรบั ณฑิ ต สาขา.pdf
08/30/2024  09:48 AM          514,863  รายงานองค ๑ ปี การสึ ษา ๖๕_วศ.บ.คอมพิ วเตอร์ .pdf
11/07/2024  10:30 PM          47,202  เอกสารร บรองห ้วช อโครงการ.docx
11/07/2024  07:28 PM          97,976  เอกสารร บรองห ้วช อโครงการ.pdf
11/06/2024  03:04 PM          98,711  เอกสารร บรองห ้วช อโครงการ_ระบบติ ดตามรประจำทาง .pdf
11/07/2024  10:31 PM          79,264  เอกสารร บรองห ้วช อโครงการ_เรื บไซค์ สำหรั บติ ดตามพณ ักงาน.pdf
               8 File(s)          1,481,456 bytes
               6 Dir(s)          672,708,800,512 bytes free

D:\>

```

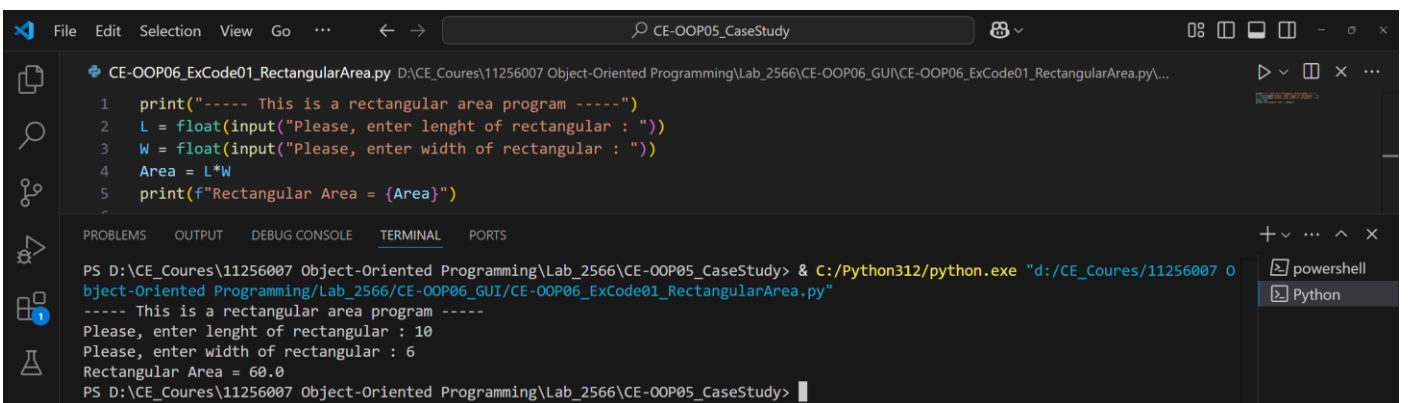
รูปที่ 1-1 การเปิดดูข้อมูลในไดรฟ์ D ด้วย Command line

จากรูปที่ 1-1 ข้อมูลที่ปรากฏให้เห็นได้บอกวันที่ใช้งานล่าสุด เวลาสุดท้ายที่เปิดใช้งาน แฟ้มข้อมูลหรือไม่ ขนาดไฟล์ และชื่อไฟล์ (DIR > Directory คือแฟ้มข้อมูล) ผู้อ่านจะเห็นว่าต้องใช้ความรู้พอประมาณสำหรับทำความเข้าใจ หากเปลี่ยนเป็นการใช้ GUI ผู้อ่านจะเข้าใจข้อมูลได้ง่ายกว่าเดิม ดังรูปที่ 1-2



รูปที่ 1-2 การเปิดดูข้อมูลในไดรฟ์ D ด้วยการ GUI

ต่อจากนี้ผู้เขียนจะกล่าวถึงการใช้งานโปรแกรมภาษา Python โดยตรงเปรียบเทียบกับการใช้งาน GUI ด้วยภาษา Python เพื่อให้ผู้อ่านได้เห็นการใช้งานที่ลำบากกับสะดวกสบาย ดังรูปที่ 1-3 คือโปรแกรมสำหรับคำนวณพื้นที่สี่เหลี่ยม



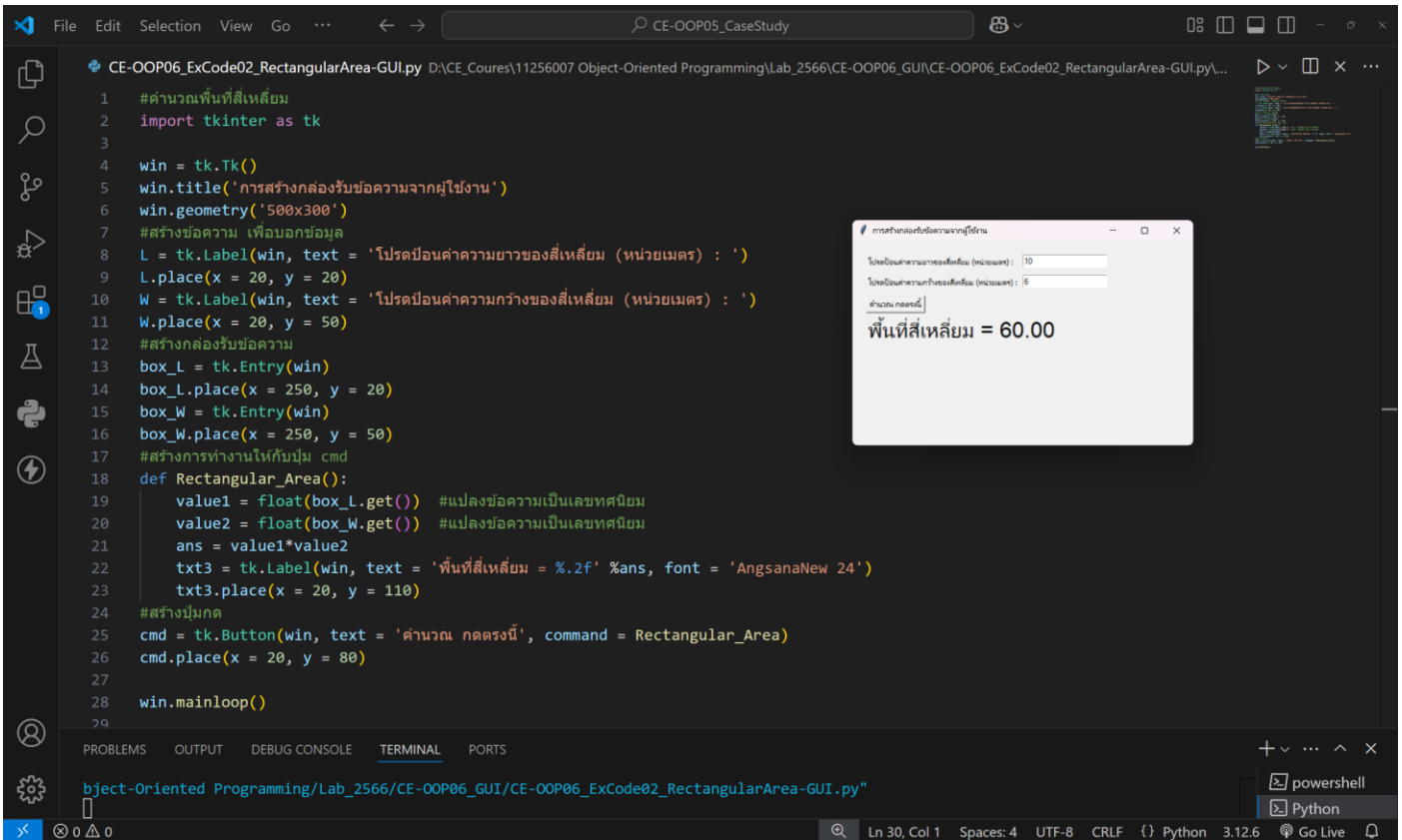
รูปที่ 1-3 การคำนวณพื้นที่สี่เหลี่ยมด้วยโค้ดภาษา Python

จากรูปที่ 1-3 ผู้อ่านจะเห็นได้ว่ามีทั้งซอร์สโค้ด (ในส่วนสำหรับเขียนโค้ด) และการใช้งานโค้ด (ในส่วนของ Terminal) ซึ่งต้องป้อนค่าตัวเลข 10 กับ 6 และได้คำตอบเป็น 60 ผู้ใช้งานทั่วไปไม่สามารถเข้าใจและใช้งานได้ แต่ผู้อ่านได้เรียนรู้การเขียนโค้ดภาษา Python บนโปรแกรม VS Code แล้ว ผู้อ่านเข้าใจและสั่งให้โค้ดทำงานได้อย่างสบาย

จากนี้ผู้เขียนจะปรับปรุงโค้ดของรูปที่ 1-3 ให้กลายเป็น GUI ผู้เขียนอยากให้ผู้อ่านได้เข้าใจและรับรู้ก่อนว่า การเปลี่ยนโค้ดที่เขียนอย่างคุ้นเคยให้กลายเป็นการเขียนโค้ดเพื่อสร้าง GUI จะมีความซับซ้อนเพิ่มขึ้น มีหลายคำสั่งที่เคยเขียนอย่างคุ้นเคยจะต้องเปลี่ยนให้เหมาะกับสร้าง GUI ผู้อ่านลองนึกถึงการแปลภาษาไทยที่คุ้นเคยให้กลายเป็นภาษาอังกฤษ มันต้องมีการแก้ไขคำและไวยากรณ์บ้าง คิดอย่างนี้จะได้เข้าใจง่ายมากขึ้น

มาถึงตรงนี้ผู้อ่านจะมีความเข้าใจมากขึ้นและเห็นภาพมากขึ้นแล้ว ต่อจากนี้ผู้เขียนจะนำเสนอคำสั่งที่เกี่ยวข้องกับการสร้าง GUI ซึ่งผู้อ่านจะได้พอเจอบ่อยมาก บ่อย ค่อนข้างบ่อย เจอเป็นระยะ และพบเจอไม่บ่อย ทั้งหมดที่ว่ามัล้วนคู่ควรให้ผู้อ่านได้เรียนรู้และนำมาใช้งาน

ให้ผู้อ่านลองดูโค้ดรูปที่ 1-4 ซึ่งเป็นโค้ดที่สร้าง GUI สำหรับคำนวณพื้นที่สี่เหลี่ยมเหมือนกับรูปที่ 1-3



รูปที่ 1-4 การคำนวณพื้นที่สี่เหลี่ยมด้วยโค้ดภาษา Python

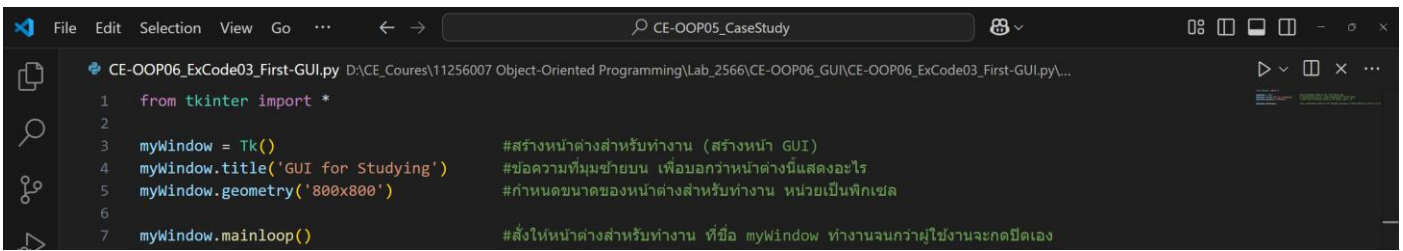
จากรูปที่ 1-4 ผู้อ่านจะพบทั้งซอร์สโค้ดสำหรับสร้าง GUI และ GUI สำหรับใช้คำนวณพื้นที่สี่เหลี่ยม ส่วนของซอร์สโค้ดมีการสร้างข้อความ สร้างช่องรับข้อมูล ปุ่มกด และการแสดงคำตอบ ซึ่งโค้ดมีความยาวและมีขั้นตอนมากกว่าโค้ดแบบที่คุ้นเคย ต่อจากนี้ผู้อ่านจะได้เรียนรู้วิธีเขียนโค้ดสร้าง GUI ด้วยภาษา Python

อย่างแรกสำหรับการเรียนรู้ GUI นั่นคือ มองทุกอย่างให้เป็นวัตถุชิ้นหนึ่ง วัตถุชิ้นนี้ต้องกำหนดให้เป็นตัวแปร (ตัวแปรต้องมีการตั้งชื่อ) อย่างที่สองเมื่อสร้างตัวแปรเสร็จแล้ว ต้องใช้เมธอดให้เหมาะกับการสร้างวัตถุตัวนี้ (ตัวแปรตัวนี้) โปรดจดจำ 2 อย่างนี้ไว้ให้ดี เมื่อผู้อ่านต้องการเขียนโค้ดสร้าง GUI

โปรดศึกษาคำสั่งต่างๆ ได้จากตอนที่ 2 ถึงตอนที่ 4

ตอนที่ 2 พื้นฐานการสร้าง GUI

1. ให้ผู้อ่านเขียนโค้ดตามตัวอย่างในรูปที่ 2-1 ซึ่งเป็นโค้ดสำหรับสร้างหน้าต่าง 1 หน้าเพื่อใช้เป็นหน้าต่างสำหรับทำงาน (GUI)



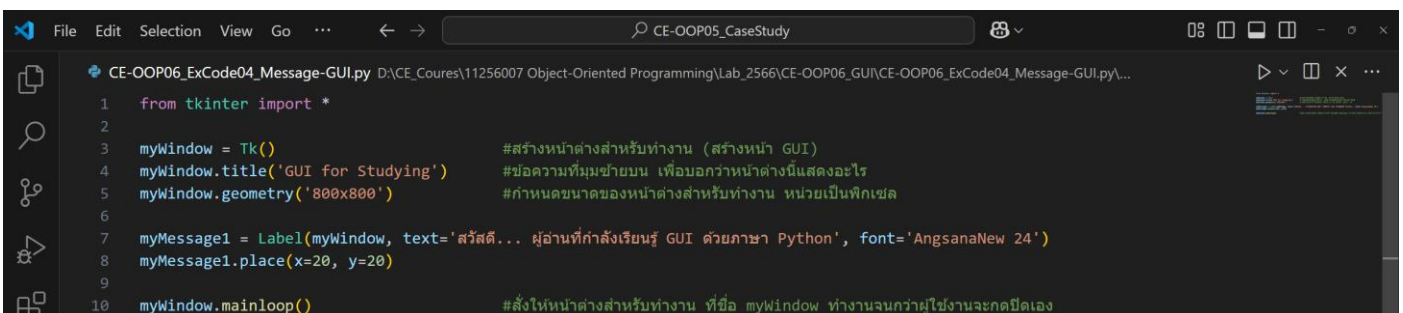
```

1  from tkinter import *
2
3  mywindow = Tk()                #สร้างหน้าต่างสำหรับทำงาน (สร้างหน้าต่าง GUI)
4  mywindow.title('GUI for Studying') #ข้อความที่มขึ้นมาบน เพื่อบอกว่าหน้าต่างนี้แสดงอะไร
5  mywindow.geometry('800x800')    #กำหนดขนาดของหน้าต่างสำหรับทำงาน หน่วยเป็นพิกเซล
6
7  mywindow.mainloop()           #สั่งให้หน้าต่างสำหรับทำงาน ที่ชื่อ mywindow ทำงานจนกว่าผู้ใช้งานจะกดปิดเอง

```

รูปที่ 2-1 การเริ่มต้นสร้างหน้าต่างสำหรับทำงาน (First GUI)

2. ผลการรันโปรแกรมเป็นอย่างไร
3. ให้ผู้อ่านเขียนโค้ดตามตัวอย่างในรูปที่ 2-2 ซึ่งเป็นการนำโค้ดรูปที่ 2-1 มาพัฒนาต่อ โดยเพิ่มการแสดงข้อความเข้าไป



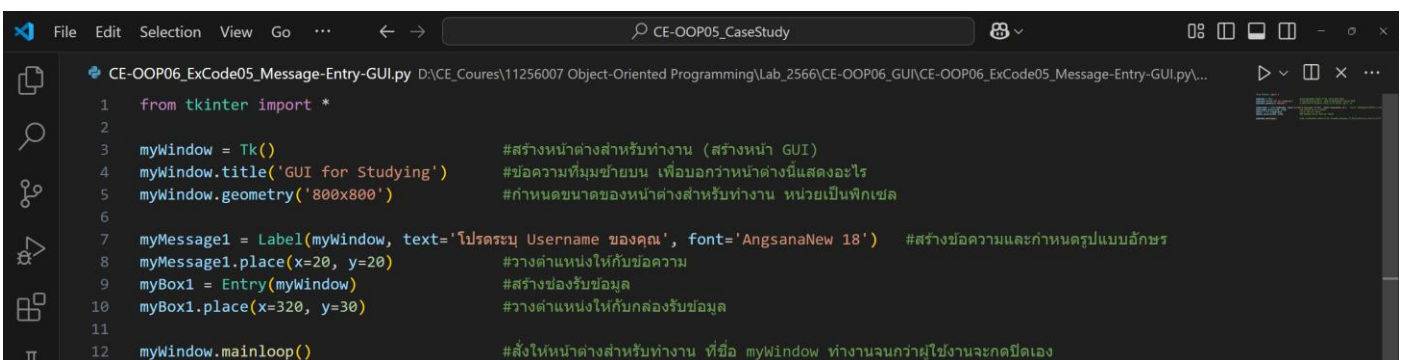
```

1  from tkinter import *
2
3  mywindow = Tk()                #สร้างหน้าต่างสำหรับทำงาน (สร้างหน้าต่าง GUI)
4  mywindow.title('GUI for Studying') #ข้อความที่มขึ้นมาบน เพื่อบอกว่าหน้าต่างนี้แสดงอะไร
5  mywindow.geometry('800x800')    #กำหนดขนาดของหน้าต่างสำหรับทำงาน หน่วยเป็นพิกเซล
6
7  myMessage1 = Label(mywindow, text='สวัสดี... ผู้อ่านที่กำลังเรียนรู้ GUI ด้วยภาษา Python', font='AngsanaNew 24')
8  myMessage1.place(x=20, y=20)
9
10 mywindow.mainloop()           #สั่งให้หน้าต่างสำหรับทำงาน ที่ชื่อ mywindow ทำงานจนกว่าผู้ใช้งานจะกดปิดเอง

```

รูปที่ 2-2 การสร้างข้อความบนหน้าต่างสำหรับทำงาน

4. ผลการรันโปรแกรมเป็นอย่างไร
5. โปรดอธิบายโค้ดบรรทัดที่ 7-8 มาให้เข้าใจ
6. ให้ผู้อ่านเขียนโค้ดตามตัวอย่างในรูปที่ 2-3 ซึ่งเป็นการนำโค้ดรูปที่ 2-2 มาพัฒนาต่อ โดยเพิ่มช่องรับข้อมูลเข้าไป



```

1  from tkinter import *
2
3  mywindow = Tk()                #สร้างหน้าต่างสำหรับทำงาน (สร้างหน้าต่าง GUI)
4  mywindow.title('GUI for Studying') #ข้อความที่มขึ้นมาบน เพื่อบอกว่าหน้าต่างนี้แสดงอะไร
5  mywindow.geometry('800x800')    #กำหนดขนาดของหน้าต่างสำหรับทำงาน หน่วยเป็นพิกเซล
6
7  myMessage1 = Label(mywindow, text='โปรดระบุ Username ของคุณ', font='AngsanaNew 18') #สร้างข้อความและกำหนดรูปแบบอักษร
8  myMessage1.place(x=20, y=20)      #วางตำแหน่งให้กับข้อความ
9  myBox1 = Entry(mywindow)         #สร้างช่องรับข้อมูล
10 myBox1.place(x=320, y=30)        #วางตำแหน่งให้กับกล่องรับข้อมูล
11
12 mywindow.mainloop()           #สั่งให้หน้าต่างสำหรับทำงาน ที่ชื่อ mywindow ทำงานจนกว่าผู้ใช้งานจะกดปิดเอง

```

รูปที่ 2-3 การสร้างข้อความและช่องรับข้อมูลบนหน้าต่างสำหรับทำงาน

7. ผลการรันโปรแกรมเป็นอย่างไร
8. คำสั่ง Entry มีไว้เพื่ออะไร
9. บรรทัดที่ 9 Entry(myWindow) คำสั่งนี้หมายความว่าอย่างไร โปรดอธิบายให้เข้าใจ
10. ให้ผู้อ่านเขียนโค้ดตามรูปที่ 2-4 ซึ่งเป็นโค้ดที่มีการรับข้อมูล สร้างปุ่ม สร้างฟังก์ชัน (เมื่อกดปุ่มแล้วให้ไปทำงานที่ฟังก์ชัน)

```

1  import tkinter as tk
2
3  win = tk.Tk()
4  win.title('การสร้างกล่องรับข้อความจากผู้ใช้งาน')
5  win.geometry('500x300')
6  #สร้างข้อความ เพื่อบอกข้อมูล
7  L = tk.Label(win, text = 'โปรดป้อนค่าความยาวของสี่เหลี่ยม (หน่วยเมตร) : ')
8  L.place(x = 20, y = 20)
9  W = tk.Label(win, text = 'โปรดป้อนค่าความกว้างของสี่เหลี่ยม (หน่วยเมตร) : ')
10 W.place(x = 20, y = 50)
11 #สร้างกล่องรับข้อความ
12 box_L = tk.Entry(win)
13 box_L.place(x = 250, y = 20)
14 box_W = tk.Entry(win)
15 box_W.place(x = 250, y = 50)
16 #สร้างการทำงานให้ปุ่ม cmd
17 def Rectangular_Area():
18     value1 = float(box_L.get()) #แปลงข้อความ เป็นเลขทศนิยม
19     value2 = float(box_W.get()) #แปลงข้อความ เป็นเลขทศนิยม
20     ans = value1*value2
21     txt3 = tk.Label(win, text = 'พื้นที่สี่เหลี่ยม = %.2f' %ans, font = 'AngsanaNew 24')
22     txt3.place(x = 20, y = 110)
23 #สร้างปุ่มกด
24 cmd = tk.Button(win, text = 'คำนวณ กดตรงนี้', command = Rectangular_Area)
25 cmd.place(x = 20, y = 80)
26
27 win.mainloop()
28
29

```

รูปที่ 2-4 การสร้าง GUI สำหรับคำนวณพื้นที่สี่เหลี่ยม

11. เมื่อรันโปรแกรมแล้ว ผู้อ่านต้องป้อนข้อมูลเป็นตัวเลขเข้าไปในช่องรับข้อมูล ผลการรันโปรแกรมเป็นอย่างไร
12. ให้ผู้อ่านเขียนโค้ดตามรูปที่ 2-5 ซึ่งเป็นโค้ดที่ใช้ Check Button เพื่อเลือกการกระทำได้หลายอย่าง


```

1  from tkinter import *
2  from tkinter import IntVar
3
4  myWindow = Tk()
5  myWindow.title('ช่องทางรับข่าวสาร')
6  myWindow.geometry('1024x720')
7  myMessage1 = Label(myWindow, text='โปรดระบุชื่อ-นามสกุลของคุณ', font='AngsanaNew 16')
8  myMessage1.place(x=20, y=30)
9  myDataBox1 = Entry(myWindow)
10 myDataBox1.place(x=280, y=38)
11 myMessage2 = Label(myWindow, text='คุณต้องการสมัครสาขาใด', font='AngsanaNew 16')
12 myMessage2.place(x=20, y=60)
13 myDataBox2 = Entry(myWindow)
14 myDataBox2.place(x=280, y=68)
15
16 def DisplayInfo():
17     userName = myDataBox1.get()
18     text1 = Label(myWindow, text='ชื่อ-นามสกุล ของคุณคือ '+userName, font='Angsana 14')
19     text1.place(x=20, y=300)
20     userMajor = myDataBox2.get()
21     text2 = Label(myWindow, text='สาขาที่คุณสมัครคือ '+userMajor, font='Angsana 14')
22     text2.place(x=20, y=330)
23     text9 = Label(myWindow, text='คุณทราบข่าวรับสมัครจากช่องทางดังต่อไปนี้', font='Angsana 14')
24     text9.place(x=20, y=360)
25     if(myCheckBox1.var.get() == 1):
26         text4 = Label(myWindow, text='Facebook')
27         text4.place(x=20, y=390)
28     if(myCheckBox2.var.get() == 1):
29         text5 = Label(myWindow, text='Instagram')
30         text5.place(x=20, y=420)
31     if(myCheckBox3.var.get() == 1):
32         text6 = Label(myWindow, text='TikTok')
33         text6.place(x=20, y=450)
34     if(myCheckBox4.var.get() == 1):
35         text7 = Label(myWindow, text='Website')
36         text7.place(x=20, y=480)
37     if(myCheckBox5.var.get() == 1):
38         text8 = Label(myWindow, text='Instagram')
39         text8.place(x=20, y=510)
40
41 myCheck1 = IntVar(); myCheck2 = IntVar(); myCheck3 = IntVar(); myCheck4 = IntVar(); myCheck5 = IntVar()
42 myMessage3 = Label(myWindow, text='คุณทราบข่าวการรับสมัครจากใดบ้าง', font='AngsanaNew 16')
43 myMessage3.place(x=20, y=110)
44
45 myCheckBox1 = Checkbutton(myWindow, text='Facebook', variable=myCheck1, onvalue=1, offvalue=0, command=DisplayInfo)
46 myCheckBox1.place(x=20, y=150); myCheckBox1.var = myCheck1
47 myCheckBox2 = Checkbutton(myWindow, text='Instagram', variable=myCheck2, onvalue=1, offvalue=0, command=DisplayInfo)
48 myCheckBox2.place(x=20, y=180); myCheckBox2.var = myCheck2
49 myCheckBox3 = Checkbutton(myWindow, text='TikTok', variable=myCheck3, onvalue=1, offvalue=0, command=DisplayInfo)
50 myCheckBox3.place(x=20, y=210); myCheckBox3.var = myCheck3
51 myCheckBox4 = Checkbutton(myWindow, text='Website', variable=myCheck4, onvalue=1, offvalue=0, command=DisplayInfo)
52 myCheckBox4.place(x=20, y=240); myCheckBox4.var = myCheck4
53 myCheckBox5 = Checkbutton(myWindow, text='Teacher', variable=myCheck5, onvalue=1, offvalue=0, command=DisplayInfo)
54 myCheckBox5.place(x=20, y=270); myCheckBox5.var = myCheck5
55
56 myWindow.mainloop()

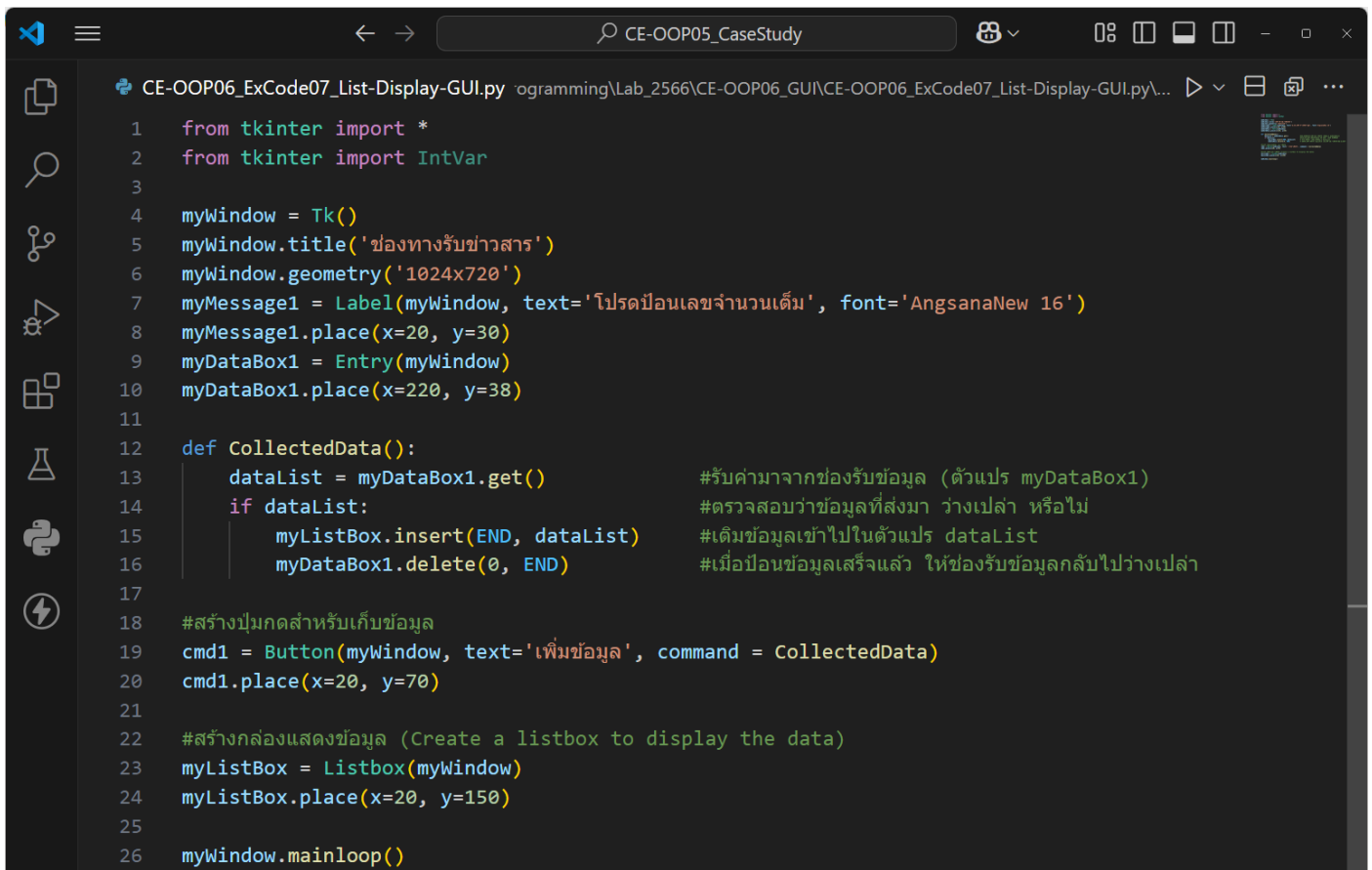
```

รูปที่ 2-4 การสร้าง GUI สำหรับคำนวณพื้นที่สี่เหลี่ยม

13. ผลลัพธ์การรันโปรแกรมเป็นอย่างไร (ขอผลลัพธ์ 2 ครั้งที่ใส่ข้อมูลไม่เหมือนกัน)
14. จากโค้ดรูปที่ 2-4 มีโค้ดอยู่บรรทัดหนึ่งที่แสดงข้อความไม่ถูกต้อง นั่นคือโค้ดบรรทัดที่เท่าไร และต้องแก้ข้อความจากอะไรให้เป็นอะไรถึงจะถูกต้อง
15. โปรดอธิบายโค้ดบรรทัดที่ 41 บรรทัดที่ 45 กับ 46 มาให้เข้าใจ

ตอนที่ 3 การประยุกต์ใช้ GUI กับสถิติ

16. ให้ผู้อ่านเขียนโค้ดตามรูปที่ 3-1 ซึ่งเป็นการป้อนข้อมูลเก็บไว้ในตัวแปรแบบ List และนำข้อมูลออกมาแสดง



```

1  from tkinter import *
2  from tkinter import IntVar
3
4  myWindow = Tk()
5  myWindow.title('ช่องทางรับข่าวสาร')
6  myWindow.geometry('1024x720')
7  myMessage1 = Label(myWindow, text='โปรดป้อนเลขจำนวนเต็ม', font='AngsanaNew 16')
8  myMessage1.place(x=20, y=30)
9  myDataBox1 = Entry(myWindow)
10 myDataBox1.place(x=220, y=38)
11
12 def CollectedData():
13     dataList = myDataBox1.get()           #รับค่ามาจากช่องรับข้อมูล (ตัวแปร myDataBox1)
14     if dataList:                         #ตรวจสอบว่าข้อมูลที่ส่งมาว่างเปล่าหรือไม่
15         myListBox.insert(END, dataList)  #เติมข้อมูลเข้าไปในตัวแปร dataList
16         myDataBox1.delete(0, END)        #เมื่อป้อนข้อมูลเสร็จแล้ว ให้ช่องรับข้อมูลกลับไปว่างเปล่า
17
18 #สร้างปุ่มกดสำหรับเก็บข้อมูล
19 cmd1 = Button(myWindow, text='เพิ่มข้อมูล', command = CollectedData)
20 cmd1.place(x=20, y=70)
21
22 #สร้างกล่องแสดงข้อมูล (Create a listbox to display the data)
23 myListBox = Listbox(myWindow)
24 myListBox.place(x=20, y=150)
25
26 myWindow.mainloop()

```

รูปที่ 3-1 การสร้าง GUI สำหรับเก็บข้อมูลไว้ในตัวแปรแบบ List

17. ให้ผู้อ่านรันโปรแกรมและลองป้อนตัวเลขเข้าไปสัก 3 ค่า ผลลัพธ์เป็นอย่างไร

18. ผู้เขียนจะนำโค้ดรูปที่ 3-1 มาพัฒนาเพิ่ม โดยมีเจตนาทำให้หาค่าผลรวมได้ ขอให้ผู้อ่านเขียนโค้ดตามรูปที่ 3-2

```

1  from tkinter import *
2  from tkinter import IntVar
3
4  #สร้างฟังก์ชันสำหรับเก็บข้อมูล (Function to add data)
5  def add_data():
6      data = myBox1.get()
7      if data:
8          value = float(data)
9          items.append(value)
10         displayBox.insert(END, data)
11         myBox1.delete(0, END)
12         mySummation()
13 #สร้างฟังก์ชันสำหรับหาค่าผลรวม (Function to summation display)
14 def mySummation():
15     total_sum = sum(items)
16     sum_label.config(text=f'Summation = {total_sum}')
17
18 #สร้างหน้าต่างสำหรับทำงาน (Create the main window)
19 win = Tk()
20 win.title('การทำงานเกี่ยวกับสถิติ')
21 win.geometry('1024x720')
22
23 #สร้างข้อความเพื่อนอกสิ่งที่ต้องทำ (Create the message for information)
24 myText1 = Label(win, text='กรุณาป้อนข้อมูลเป็นตัวเลข : ', font='AngsanaNew 16')
25 myText1.place(x=20, y=20)
26
27 #สร้างช่องรับข้อมูล (Create an entry widget for input)
28 myBox1 = Entry(win)
29 myBox1.place(x=270, y=30)
30
31 #สร้างปุ่มกดเพื่อเก็บข้อมูล
32 myButton1 = Button(win, text='ป้อนข้อมูล', command=add_data)
33 myButton1.place(x=20, y=60)
34
35 #สร้างกล่องแสดงข้อมูลใน List (Create a listbox to display the data)
36 displayBox = Listbox(win)
37 displayBox.place(x=20, y=300)
38
39 #สร้างข้อความเพื่อแสดงค่าผลรวม (Create a label to display summation)
40 sum_label = Label(win, text='Summation = 0.00')
41 sum_label.place(x=20, y=200)
42
43 #สร้างตัวแปร List เพื่อเอาไว้เก็บข้อมูล (Initialize an empty list to store data)
44 items = []
45
46 #โปรแกรมทำงาน (Run the application)
47 win.mainloop()

```

รูปที่ 3-2 การสร้าง GUI สำหรับเก็บข้อมูลไว้ในตัวแปรแบบ List และหาค่าผลรวม

19. ผลการรันโปรแกรมเป็นอย่างไร (ให้ผู้อ่านป้อนตัวเลขเข้าไป 10 ค่า)

Checkpoint1 ให้ผู้อ่านเพิ่มการหาค่าเฉลี่ย หาค่าสูงสุด หาค่าต่ำสุด ค่าความแปรปรวน และค่าเบี่ยงเบนมาตรฐาน เมื่อเขียนโค้ดเสร็จแล้ว โปรแกรมแสดงผลลัพธ์และโค้ดให้เห็นอย่างชัดเจน (ป้อนตัวเลขสัก 10 ค่า)

Checkpoint2 จากโค้ดรูปที่ 3-2 การหาค่าผลรวมได้เกิดขึ้นทันทีหลังจากการป้อนตัวเลขเข้าไป ให้ผู้อ่านแก้ไขโค้ดเป็น กดปุ่มเพื่อหาค่าผลรวม เมื่อกดปุ่มค่าผลรวมจึงแสดงผล ต่อจากนั้นเพิ่มปุ่มหาค่าเฉลี่ย ค่าสูงสุด ค่าต่ำสุด ค่าความแปรปรวน และค่าเบี่ยงเบนมาตรฐาน เมื่อกดปุ่มเหล่านี้ค่าที่ต้องการจะแสดงออกบน GUI ที่สร้างไว้ โปรแกรมโค้ดและผลลัพธ์ให้ชัดเจน