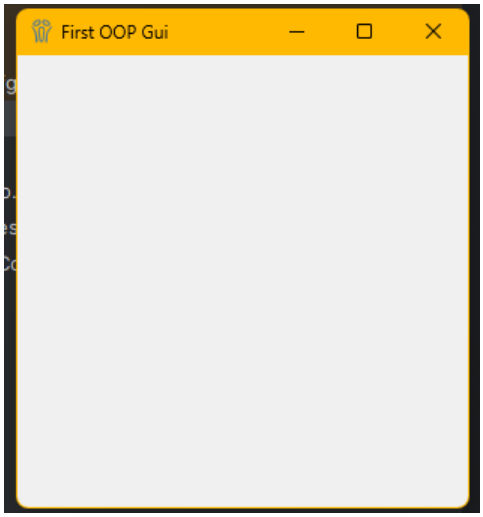
Activity No. 4.1	
Stacks	
Course Code: CPE009B	**Program:** Computer Engineering
Course Title: Object Oriented Programming	**Date Performed:** 10/14/24
Section: Cpe21s4	**Date Submitted:** 10/14/24
Name(s): Anna Marie Zolina	**Instructor:** Mrs. Maria Rizette Sayo

6. Output

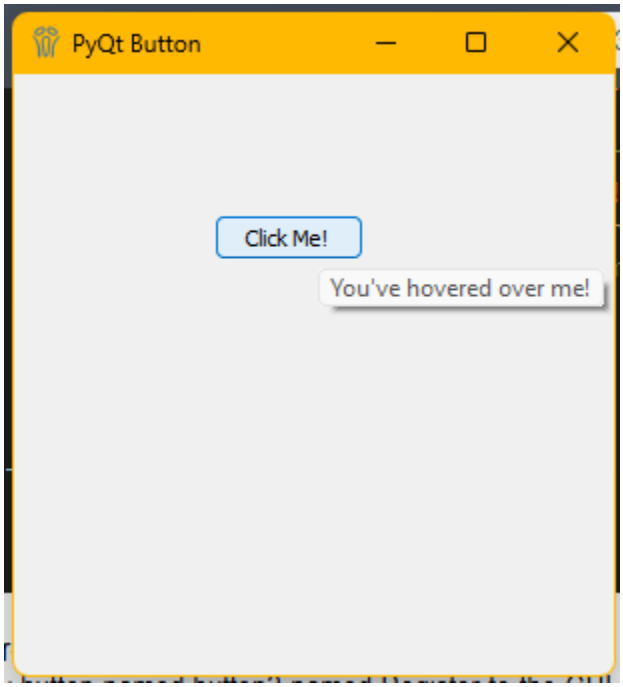
```
import sys
from PyQt5.QtWidgets import QMainWindow, QApplication
from PyQt5.QtGui import QIcon


class App(QMainWindow):

    def __init__(self):
        super().__init__()
        self.title = "First OOP Gui"
        self.initUI()

    def initUI(self):
        self.setWindowTitle(self.title)
        self.setGeometry(200, 200, 300, 300)
        self.setWindowIcon(QIcon('pythonico.ico'))
        self.show()


if __name__ == '__main__':
    app = QApplication(sys.argv)
    Main = App()
    sys.exit(app.exec_())
```

OUTPUT:



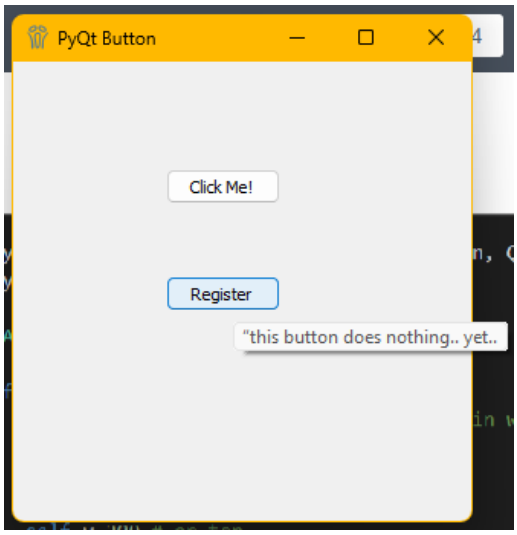
```
import sys
from PyQt5.QtWidgets import QWidget, QApplication, QMainWindow, QPushButton
from PyQt5.QtGui import QIcon


class App(QWidget):

    def __init__(self):
        super().__init__()
        self.title = "PyQt Button"
        self.x = 200
        self.y = 200
        self.width = 300
        self.height = 300
        self.initUI()

    def initUI(self):
        self.setWindowTitle(self.title)
        self.setGeometry(self.x, self.y, self.width, self.height)
        self.setWindowIcon(QIcon('pythonico.ico'))

        self.button = QPushButton('Click Me!', self)
        self.button.setToolTip("You've hovered over me!")
        self.button.move(100, 70)

        self.show()

if __name__ == '__main__':
    app = QApplication(sys.argv)
    ex = App()
    sys.exit(app.exec_())
```

OUTPUT:



Adding New Button:

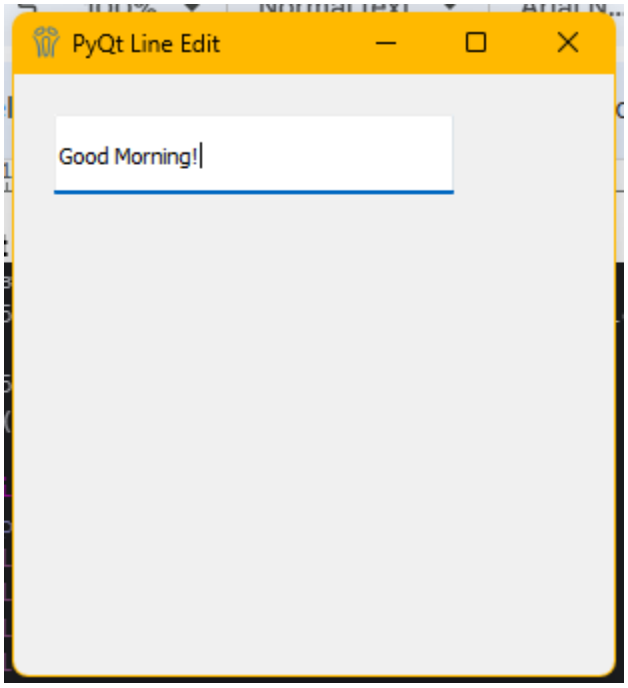
```
self.button2 = QPushButton('Click Me!', self)
self.button2.setToolTip("“this button does nothing.. yet..")
self.button2.move(100, 140)
```



Add an import QLineEdit to the Pycharm.Widgets import:

```
import sys
from PyQt5.QtWidgets import QWidget, QApplication, QMainWindow, QPushButton,
QLineEdit
from PyQt5.QtGui import QIcon
class App(QWidget):

    def __init__(self):
        super().__init__()
        self.title= "PyQt Line Edit"
        self.x = 200
        self.y = 200
        self.width = 300
        self.height = 300
        self.initUI()

    def initUI(self):
        self.setWindowTitle(self.title)
        self.setGeometry(self.x, self.y, self.width, self.height)
        self.setWindowIcon(QIcon('pythonico.ico'))

        self.textbox = QLineEdit(self)
        self.textbox.move(20, 20)
        self.textbox.resize(200, 40)

        self.show()

if __name__ == '__main__':
    app = QApplication(sys.argv)
    ex = App()
    sys.exit(app.exec_())
```

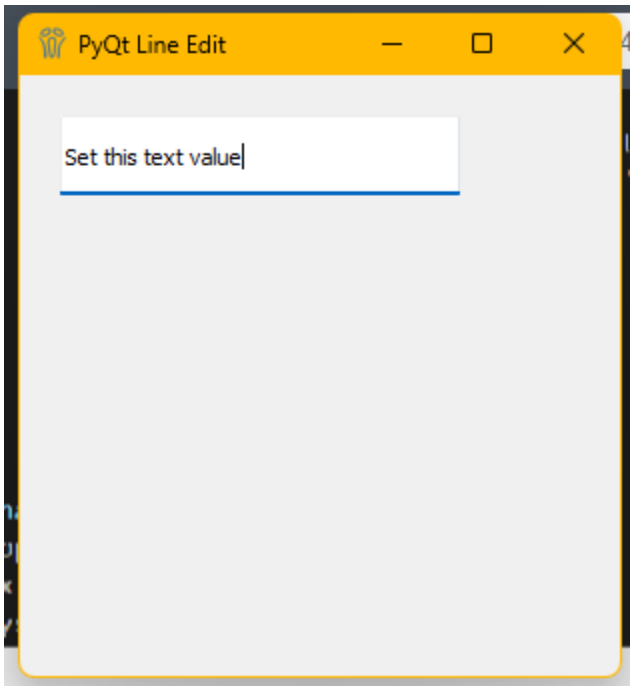
OUTPUT:



Add the following code below self.textbox.resize():

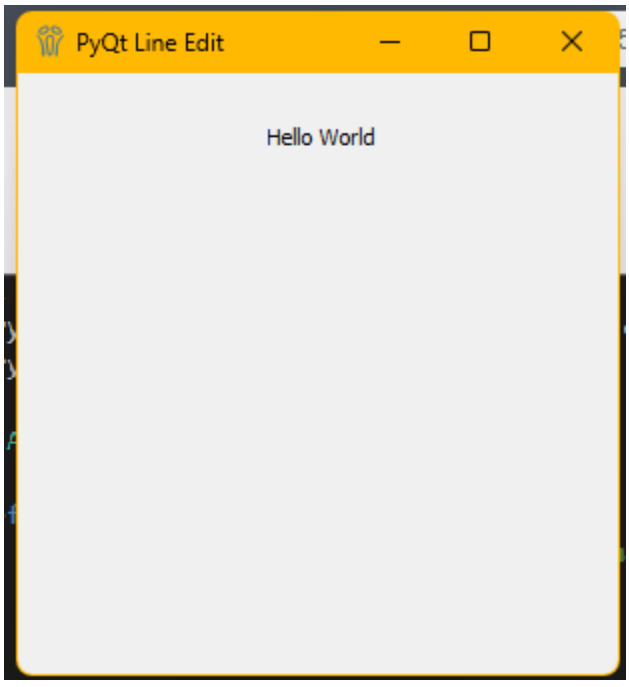
```
self.textbox.setText("Set this text value")
```

OUTPUT:



```
import sys
from PyQt5.QtWidgets import QWidget, QApplication, QMainWindow, QPushButton,
QLineEdit, QLabel
from PyQt5.QtGui import QIcon

class App(QWidget):

   def __init__(self):
       super().__init__()
       self.title = "PyQt Line Edit"
       self.x = 200
       self.y = 200
       self.width = 300
       self.height = 300
       self.initUI()

   def initUI(self):
       self.setWindowTitle(self.title)
       self.setGeometry(self.x, self.y, self.width, self.height)
       self.setWindowIcon(QIcon('pythonico.ico'))

       self.textboxlbl = QLabel("Hello World", self)
       self.textboxlbl.move(125, 25)

       self.show()

if __name__ == '__main__':
   app = QApplication(sys.argv)
   ex = App()
   sys.exit(app.exec_())
```

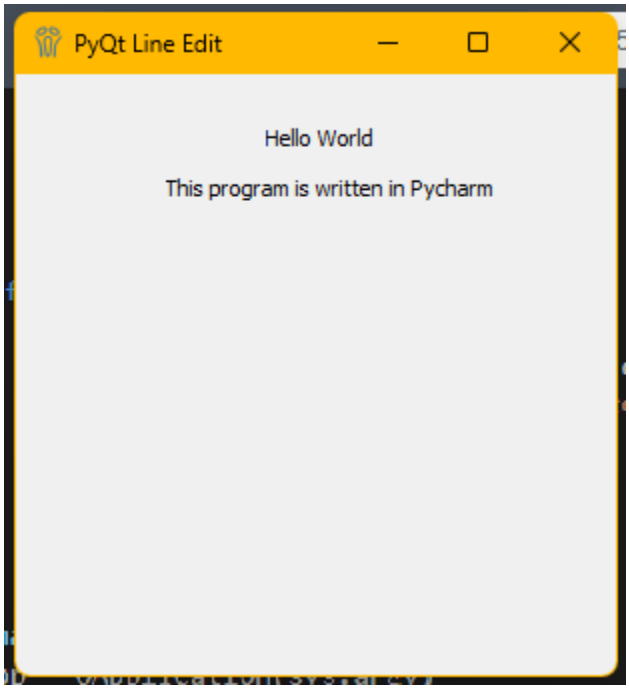
OUTPUT:



Create a new label called “This program is written in Pycharm” and place it at the center and below the Hello World!:

```
self.textboxlbl2 = QLabel("This program is written in Pycharm", self)
self.textboxlbl2.move(75, 50)
```

OUTPUT:



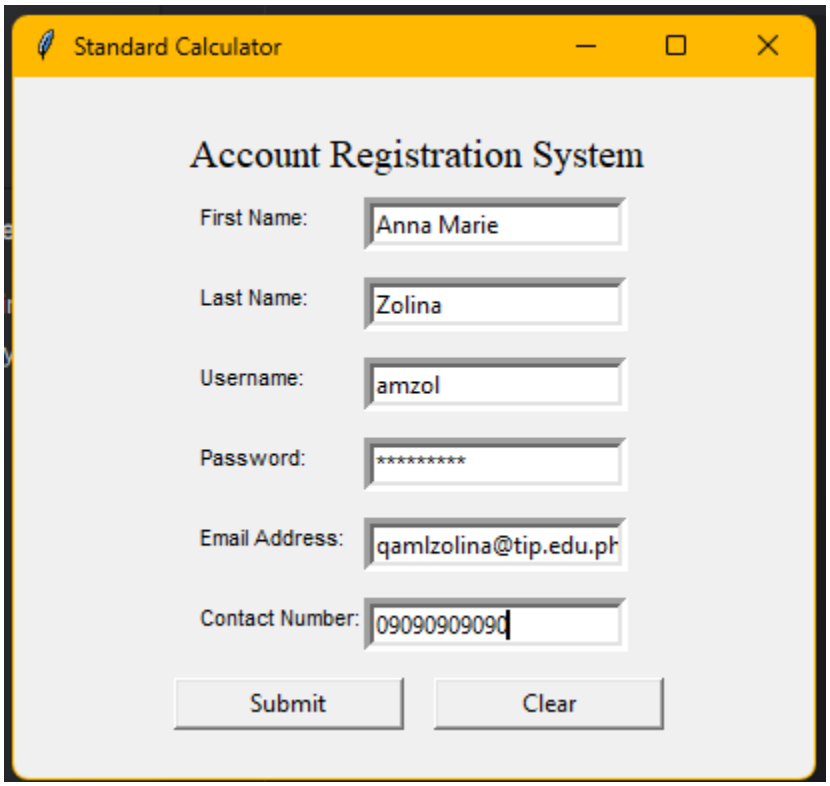
7. Supplementary Activity

```
from tkinter import *
from tkinter import messagebox

class MyWindow:
   def __init__(self, win):

       self.Label1 = Label(win, fg="BLACK", text = "Account Registration
System", font = ("Times New Roman",15))
       self.Label1.place(x=80, y=25)

       self.Label2 = Label(win, fg="BLACK", text = "First Name: ", font =
("Arial",8,))
       self.Label2.place(x=90, y=60)

       self.Entry1 = Entry(win, bd=5)
       self.Entry1.place(x=175, y=60)

       self.Label3 = Label(win, fg="BLACK", text="Last Name: ", font =
("Arial",8))
       self.Label3.place(x=90, y=100)

       self.Entry2 = Entry(win, bd=5)
       self.Entry2.place(x=175, y=100)

       self.Label4 = Label(win, fg="BLACK", text="Username: ", font =
("Arial",8))
       self.Label4.place(x=90, y=140)
```

```
        self.Entry3 = Entry(win, bd=5)
        self.Entry3.place(x=175, y=140)

        self.Label5 = Label(win, fg="BLACK", text="Password: ", font=("Arial",
8))
        self.Label5.place(x=90, y=180)

        self.Entry4 = Entry(win, bd=5, show = '*')
        self.Entry4.place(x=175, y=180)

        self.Label6 = Label(win, fg="BLACK", text="Email Address: ",
font=("Arial", 8))
        self.Label6.place(x=90, y=220)

        self.Entry5 = Entry(win, bd=5)
        self.Entry5.place(x=175, y=220)

        self.Label7 = Label(win, fg="BLACK", text="Contact Number: ",
font=("Arial", 8))
        self.Label7.place(x=90, y=260)

        self.Entry6 = Entry(win, bd=5)
        self.Entry6.place(x=175, y=260)

        self.Button = Button(win, fg="BLACK",height =1, width = 15,
text="Clear", command=self.clear)
        self.Button.place(x=210, y=300)

        self.Button2 = Button(win, fg="BLACK", height=1, width=15,
text="Submit", command=self.submit)
        self.Button2.place(x=80, y=300)


    def submit(self):
        messagebox.showinfo('Account Registration', 'Your Account Has Been
Verified!')
    def clear(self):
        self.Entry1.delete(0, 'end')
        self.Entry2.delete(0, 'end')
        self.Entry3.delete(0, 'end')
        self.Entry4.delete(0, 'end')
        self.Entry5.delete(0, 'end')
        self.Entry6.delete(0, 'end')
```

```
from registration import MyWindow
from tkinter import *

window = Tk()
MyWin=MyWindow(window)
window.geometry("400x350+10+10")
window.title("Standard Calculator")
window.mainloop()
```

OUTPUT:



Questions

1. What are the common GUI Applications that general end-users such as home users, students, and office employees use? (give at least 3 and describe each)

- GUI applications, such as web browsers and office suites, demonstrate how graphical interfaces facilitate user interactions. Your account registration system will also allow users to enter information using intuitive elements such as buttons and text fields.

2. Based from your answer in question 1, why do you think home users, students, and office employees use those GUI programs?

- By including clearly labeled text fields next to their inputs, your registration system will exemplify the organization and ease of use that users prefer in GUI programs. Users will be encouraged to finish the registration process without experiencing any frustration thanks to this user-friendly design.

3. How does Pycharm help developers in making GUI applications, what would be the difference if developers made GUI programs without GUI Frameworks such as Pycharm or Tkinter?

- By offering tools like code completion and debugging features that make creating GUI components easier, PyCharm improves your development of the registration system. This frees you from worrying about the intricacies of GUI development so you can concentrate on creating a flawless user experience.

4. What are the different platforms a GUI program may be created and deployed on? (Three is required then state why might a program be created on that specific platform)

- Platforms like Windows, macOS, and Linux can be used to create your registration system; Windows is a popular option because of its large user base and compatibility with development tools. This guarantees that a large audience will be able to access your application.

5. What is the purpose of app = QApplication(sys.argv), ex = App(), and sys.exit(app.exec_())?

- Your GUI application is initialized, the main interface for user input is created, and the event loop for user interactions is started by the snippet app = QApplication(sys.argv), ex = App(), and sys.exit(app.exec_()). This framework is necessary to make sure your registration system functions properly and reacts to user input.

8. Conclusion

In conclusion, GUI applications including as word processors, spreadsheets, and web browsers play an important part in the daily lives of home users, students, and office workers because they provide easy interfaces for difficult tasks. Tools like PyCharm and frameworks like Tkinter make the development process easier, allowing developers to focus on functionality rather than low-level code details. With multiple deployment platforms available, developers may reach a wide range of users while adapting their programs to specific user demands. Overall, the combination of user-friendly design and sophisticated development tools is helping to define the landscape of software applications.

9. Assessment Rubric