Activity No. 4 - Introduction to GUI Development using Pycharm	
Introduction to GUI Development using Pycharm	
Course Code: CPE009B	**Program:** Computer Engineering
Course Title: Object Oriented Programming 2	**Date Performed: 10/14/24**
Section: CPE 21S4	**Date Submitted: 10/14/24**
Name(s): Dominic Joseph P. Virtucio	**Instructor: Ma'am Sayo**
Supplementary Activity	

Python

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

class RegistrationForm(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("Account Registration")
        self.setGeometry(100, 100, 340, 275)
        self.setWindowIcon(QIcon('monkey.ico'))
        self.initUI()

    def initUI(self):

        # Labels
        self.first_name_label = QLabel("First Name:", self)
        self.first_name_label.move(60, 20)
        self.last_name_label = QLabel("Last Name:", self)
        self.last_name_label.move(60, 50)
        self.username_label = QLabel("Username:", self)
        self.username_label.move(60, 80)
        self.password_label = QLabel("Password:", self)
        self.password_label.move(60, 110)
        self.email_label = QLabel("Email Address:", self)
        self.email_label.move(60, 140)
        self.contact_label = QLabel("Contact No.:", self)
        self.contact_label.move(60, 170)

        # Text Fields
        self.first_name_input = QLineEdit(self)
        self.first_name_input.move(140, 20)
        self.last_name_input = QLineEdit(self)
        self.last_name_input.move(140, 50)
        self.username_input = QLineEdit(self)
        self.username_input.move(140, 80)
        self.password_input = QLineEdit(self)
        self.password_input.move(140, 110)
        self.password_input.setEchoMode(QLineEdit.Password)
        self.email_input = QLineEdit(self)
        self.email_input.move(140, 140)
        self.contact_input = QLineEdit(self)
        self.contact_input.move(140, 170)
```

```
        # Buttons
        self.submit_button = QPushButton("Submit", self)
        self.submit_button.move(80, 210)
        self.submit_button.clicked.connect(self.show_success)

        self.clear_button = QPushButton("Clear", self)
        self.clear_button.move(180, 210)
        self.clear_button.clicked.connect(self.clear_all)

        self.show()

    def clear_all(self):
        self.first_name_input.clear()
        self.last_name_input.clear()
        self.username_input.clear()
        self.password_input.clear()
        self.email_input.clear()
        self.contact_input.clear()

    def show_success(self):
        QMessageBox.information(self, "Success", "Account successfully registered!")


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

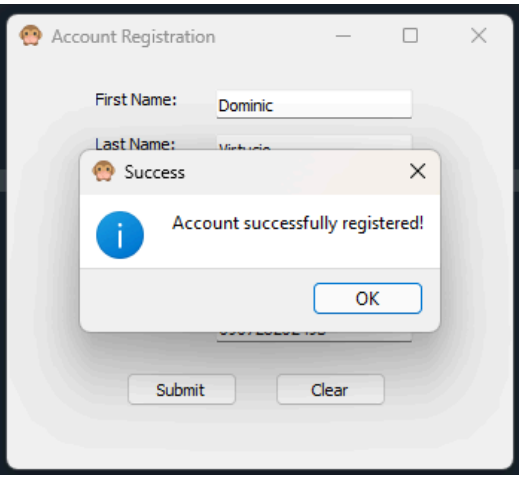
Output

Output of main.py

Output when button “Submit” is clicked

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

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



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)**

Web browsers, word processors, and email clients are a few popular GUI applications. Users can view and download files, navigate the internet, and view web sites via web browsers. Text document generation and editing are made possible by word processors, and email programs facilitate email management through message sending, receiving, and organization.

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

GUI programs are preferred by office workers, students, and home users due to its ease of use and intuitiveness. These programs use graphical user interfaces to make software interaction simpler and do away with the need to learn intricate commands. These programs facilitate efficient task completion by providing users with menus, buttons, and icons.

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

PyCharm improves the creation of GUI applications by offering a feature-rich environment that makes the process more efficient. It provides strong debugging tools, intelligent code completion, and visual design tools to make user interface creation easier. Developers would have to individually implement every GUI element in the absence of GUI frameworks, which would have led to more complex code and longer development times.

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)**

It is possible to build and implement GUI programs on Linux, macOS, and Windows platforms. Because of its vast user base, Windows is quite popular and perfect for programs that are meant for a wide audience. While Linux gives flexibility and customization choices that developers prefer, macOS is selected because of its emphasis on design and user experience.

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

Using the PyQt framework, these lines of code are essential for building and executing a GUI application. While `ex = App()} generates an instance of the main application window, `app = QApplication(sys.argv)} initializes the application and handles command-line arguments. Lastly, the application's event loop is started by {sys.exit(app.exec_())}, guaranteeing that it will respond to user inputs until it is closed.

Conclusion
Using PyCharm and the PyQt framework, this activity provided me with a hands-on introduction to GUI creation. I was able to better understand the fundamentals of GUI programming by creating a basic form for account registration using several GUI elements, such as buttons, text fields, and labels. Through the use of pre-built components, frameworks like PyQt streamline development. The experience also demonstrated the benefits of selecting the appropriate platform depending on target audience and design requirements. All in all, it demonstrated how effective these frameworks can be for producing more complicated interfaces and provided a strong basis for understanding GUI creation.
Assessment Rubric