**Lab Exercise 8 – Two Number Calculator in PyQT**

**Lab Exercise: Creating a Two-Number Calculator with PyQt**

Creating a lab exercise for building a two-number calculator using PyQt involves providing a hands-on task for learners to create a simple calculator application in Python with a graphical user interface. In this exercise, learners will build a basic calculator that can perform addition, subtraction, multiplication, and division operations on two numbers entered by the user. Here's a step-by-step lab exercise:

Objective: Create a PyQt application with a graphical user interface for performing arithmetic operations on two numbers.

**Requirements:**

* PyQt5 should be installed.
* Basic understanding of Python and PyQt5.

**Instructions:**

In this lab exercise, learners will create a two-number calculator application using PyQt. Follow the steps below:

**Create the Python Script:**

Create a Python script, e.g., calculator.py, and add the following code to create a basic PyQt application:

import sys

from PyQt5.QtWidgets import QApplication, QWidget, QVBoxLayout, QHBoxLayout, QLineEdit, QPushButton, QLabel

class CalculatorApp(QWidget):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.initUI()

def initUI(self):

# Create input fields

self.num1\_input = QLineEdit(self)

self.num2\_input = QLineEdit(self)

# Create buttons for arithmetic operations

self.add\_button = QPushButton('+', self)

self.subtract\_button = QPushButton('-', self)

self.multiply\_button = QPushButton('\*', self)

self.divide\_button = QPushButton('/', self)

# Create a label to display the result

self.result\_label = QLabel('', self)

# Set up layouts

input\_layout = QVBoxLayout()

input\_layout.addWidget(self.num1\_input)

input\_layout.addWidget(self.num2\_input)

button\_layout = QVBoxLayout()

button\_layout.addWidget(self.add\_button)

button\_layout.addWidget(self.subtract\_button)

button\_layout.addWidget(self.multiply\_button)

button\_layout.addWidget(self.divide\_button)

main\_layout = QHBoxLayout()

main\_layout.addLayout(input\_layout)

main\_layout.addLayout(button\_layout)

main\_layout.addWidget(self.result\_label)

self.setLayout(main\_layout)

# Connect button clicks to functions

self.add\_button.clicked.connect(self.add)

self.subtract\_button.clicked.connect(self.subtract)

self.multiply\_button.clicked.connect(self.multiply)

self.divide\_button.clicked.connect(self.divide)

def add(self):

num1 = float(self.num1\_input.text())

num2 = float(self.num2\_input.text())

result = num1 + num2

self.result\_label.setText(f'Result: {result}')

def subtract(self):

num1 = float(self.num1\_input.text())

num2 = float(self.num2\_input.text())

result = num1 - num2

self.result\_label.setText(f'Result: {result}')

def multiply(self):

num1 = float(self.num1\_input.text())

num2 = float(self.num2\_input.text())

result = num1 \* num2

self.result\_label.setText(f'Result: {result}')

def divide(self):

num1 = float(self.num1\_input.text())

num2 = float(self.num2\_input.text())

if num2 != 0:

result = num1 / num2

self.result\_label.setText(f'Result: {result}')

else:

self.result\_label.setText('Error: Division by zero')

if \_\_name\_\_ == '\_\_main\_\_':

app = QApplication(sys.argv)

window = CalculatorApp()

window.setWindowTitle('Two-Number Calculator')

window.show()

sys.exit(app.exec\_())

Run the Application:

Open a terminal or command prompt, navigate to the directory containing the Python script (calculator.py), and run the script:

python calculator.py

The PyQt application should open and display a simple two-number calculator.