**Lab Exercise 26- QProperty and QML Component**

Here's an example that uses QtCore.Property in PySide6 and QML, this time with a QML Label that reflects the value of the property.

**Python script (main.py):**

import sys

from PySide6.QtCore import QObject, Property, Signal, Slot

from PySide6.QtGui import QGuiApplication

from PySide6.QtQml import QQmlApplicationEngine

class MyObject(QObject):

valueChanged = Signal(int)

def \_\_init\_\_(self, parent=None):

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

self.\_value = 0

def get\_value(self):

return self.\_value

def set\_value(self, new\_value):

if self.\_value != new\_value:

self.\_value = new\_value

self.valueChanged.emit(self.\_value)

value = Property(int, get\_value, set\_value, notify=valueChanged)

if \_\_name\_\_ == "\_\_main\_\_":

app = QGuiApplication(sys.argv)

my\_object = MyObject()

engine = QQmlApplicationEngine()

engine.rootContext().setContextProperty("myObject", my\_object)

engine.load("main.qml")

if not engine.rootObjects():

sys.exit(-1)

sys.exit(app.exec())

**QML file (main.qml):**

import QtQuick

import QtQuick.Controls

ApplicationWindow {

visible: true

width: 400

height: 400

Column {

anchors.centerIn: parent

spacing: 20

Label {

id: valueLabel

text: "Value from Python: " + myObject.value

horizontalAlignment: Text.AlignHCenter

}

Button {

text: "Increment Value"

onClicked: myObject.value = myObject.value + 1

}

}

}

**Explanation:**

* The Python script creates a MyObject class that defines a property value with the necessary getter, setter, and signal.
* The QML file main.qml creates an ApplicationWindow with a Column layout. Inside the column, there's a Label that displays the value of the property and a Button that increments the property when clicked.
* The PySide6 application initializes the QML engine, sets the context property myObject to the instance of MyObject, and loads the QML file.
* Running the application will display the QML interface. Clicking the "Increment Value" button will update the displayed value, which will be reflected in the QML Label.

This example demonstrates how to use QtCore.Property in PySide6 and QML to create a simple interactive application with dynamic property updates.