Laboratory Activity 6 - GUI Design: Layout and Styling	
Bonifacio, Nyko Adrein L.	10-28-2024
CpE 009B - CpE21 S4	Prof. Sayo

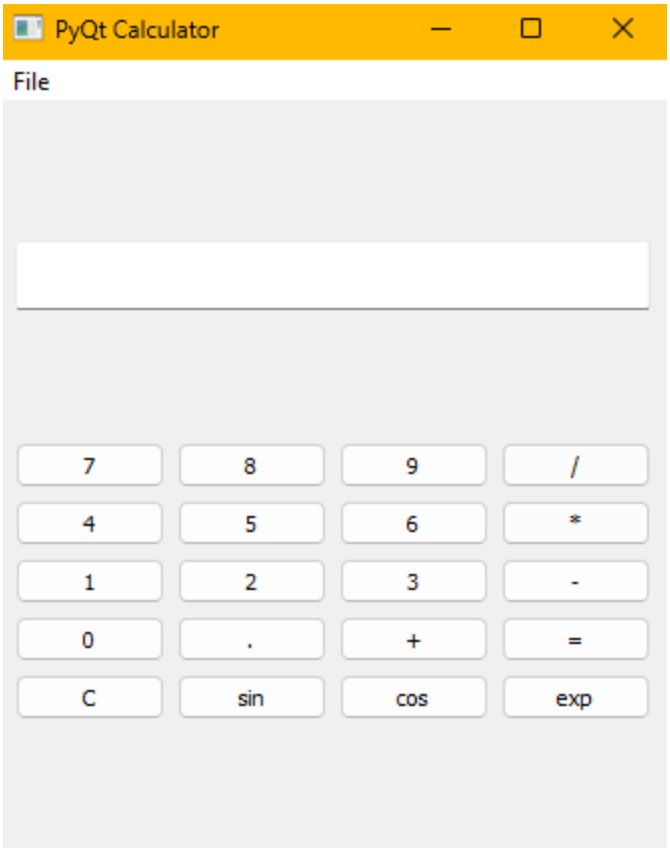
6. Supplementary Task

Python

```python
import sys
import math
from PyQt5.QtWidgets import QApplication, QMainWindow, QWidget, QVBoxLayout,
QGridLayout, QPushButton, QLineEdit, QAction, QFileDialog, QMessageBox
from PyQt5.QtGui import QKeySequence
from PyQt5.QtCore import Qt

class Calculator(QMainWindow):
    def __init__(self):
        super().__init__()

        self.setWindowTitle("PyQt Calculator")
        self.setGeometry(100, 100, 300, 400)

        self.current_expression = ""
        self.result = ""

        self.create_main_layout()
        self.create_menu_bar()

    def create_main_layout(self):
        widget = QWidget()
        main_layout = QVBoxLayout()

        self.display = QLineEdit()
        self.display.setReadOnly(True)
        self.display.setAlignment(Qt.AlignRight)
        self.display.setFixedHeight(35)
        main_layout.addWidget(self.display)

        buttons_layout = QGridLayout()

        buttons = [
            ('7', 0, 0), ('8', 0, 1), ('9', 0, 2), ('/', 0, 3),
            ('4', 1, 0), ('5', 1, 1), ('6', 1, 2), ('*', 1, 3),
            ('1', 2, 0), ('2', 2, 1), ('3', 2, 2), ('-', 2, 3),
            ('0', 3, 0), ('.', 3, 1), ('+', 3, 2), ('=', 3, 3),
            ('C', 4, 0), ('sin', 4, 1), ('cos', 4, 2), ('exp', 4, 3)
        ]
```

```
        for text, row, col in buttons:
            button = QPushButton(text)
            button.clicked.connect(self.on_button_clicked)
            buttons_layout.addWidget(button, row, col)

        main_layout.addLayout(buttons_layout)
        widget.setLayout(main_layout)
        self.setCentralWidget(widget)

    def create_menu_bar(self):
        menu_bar = self.menuBar()
        file_menu = menu_bar.addMenu("File")

        save_action = QAction("Save", self)
        save_action.triggered.connect(self.save_to_file)
        file_menu.addAction(save_action)

        load_action = QAction("Load", self)
        load_action.triggered.connect(self.load_from_file)
        file_menu.addAction(load_action)

        exit_action = QAction("Exit", self)
        exit_action.setShortcut(QKeySequence("Ctrl+Q"))
        exit_action.triggered.connect(self.close)
        file_menu.addAction(exit_action)

    def on_button_clicked(self):
        sender = self.sender()
        text = sender.text()

        if text == 'C':
            self.current_expression = ""
            self.result = ""
            self.display.setText("")
        elif text == '=':
            try:
                self.result = str(eval(self.current_expression))
                self.display.setText(self.result)
                self.current_expression = self.result
            except Exception as e:
                self.display.setText("Error")
        elif text in ('sin', 'cos', 'exp'):
            try:
                value = float(self.current_expression) if
self.current_expression else 0.0
                if text == 'sin':
                    self.result = str(math.sin(math.radians(value)))
                elif text == 'cos':
                    self.result = str(math.cos(math.radians(value)))
                elif text == 'exp':
```

```
                    self.result = str(math.exp(value))
                self.display.setText(self.result)
                self.current_expression = self.result
            except Exception as e:
                self.display.setText("Error")
        else:
            self.current_expression += text
            self.display.setText(self.current_expression)

    def save_to_file(self):
        file_name, _ = QFileDialog.getSaveFileName(self, "Save Operations",
"", "Text Files (*.txt)")
        if file_name:
            with open(file_name, 'w') as file:
                file.write(f"Expression: {self.current_expression}\n")
                file.write(f"Result: {self.result}\n")

    def load_from_file(self):
        file_name, _ = QFileDialog.getOpenFileName(self, "Load Operations",
"", "Text Files (*.txt)")
        if file_name:
            with open(file_name, 'r') as file:
                content = file.read()
                QMessageBox.information(self, "Loaded Operations", content)

if __name__ == "__main__":
    app = QApplication(sys.argv)
    window = Calculator()
    window.show()
    sys.exit(app.exec_())
```



7. Conclusion

In conclusion, GUI design's layout and styling is important for creating a user-friendly interface. A well-organized layout makes navigation easier, while thoughtful styling (like colors and fonts) makes the interface appealing and engaging. Learning about these has shown me that good design is important as the program’s functions, because it greatly affects how users feel when using the software.