Task delivery document:

OPTIMIZING THE USE OF OOP PRINCIPLES

Class Structure and Description

ExpenseMenu (inherits QMenuBar)

This class creates the menu bar with "File," "Edit," and "Help" menu options.

Purpose: Adds a basic menu structure to the app's main window, enhancing UI consistency.

ExpenseTable (inherits QTableWidget)

Sets up a table with two columns: "Expense" and "Price."

Initializes with some default expenses and has methods to:

Add a new expense to the table with a name and price.

Calculate the total of all expenses listed in the table.

Purpose: Provides a display and storage area for added expenses.

ExpenseInputPanel (inherits QWidget)

Creates input fields for users to enter an expense name and price.

Contains a button to add the inputted expense to the ExpenseTable.

Error Handling: Ensures the price entered is a valid number.

Purpose: Captures user input for new expenses and triggers the total update.

ExpenseApp (inherits QMainWindow)

The main application window that brings together the other components.

Sets up the overall layout, combining the menu bar, ExpenseTable, ExpenseInputPanel, and a display area for the total.

OOP Principles Optimized:

Encapsulation:

Each class now encapsulates its own functionality. For instance, ExpenseTable handles all table-related actions, ExpenseInputPanel deals with user input, and ExpenseMenu handles the menu setup. This reduces dependency on global variables and keeps data hidden within each class.

Single Responsibility Principle (SRP):

Each class has a single, specific responsibility:

ExpenseTable manages the table data.

ExpenseInputPanel handles input and validation.

ExpenseMenu sets up the menu bar.

ExpenseApp integrates these components into a cohesive UI.

Modularity:

The code is broken down into smaller, independent classes, making it more modular and easier to maintain or extend. If more features need to be added to the input panel or the table, they can be done within those specific classes without affecting other parts of the application.

Improved Reusability:

Classes like ExpenseTable and ExpenseInputPanel can be reused in other applications with minimal changes.