Specification for a C++ Five-Function Calculator

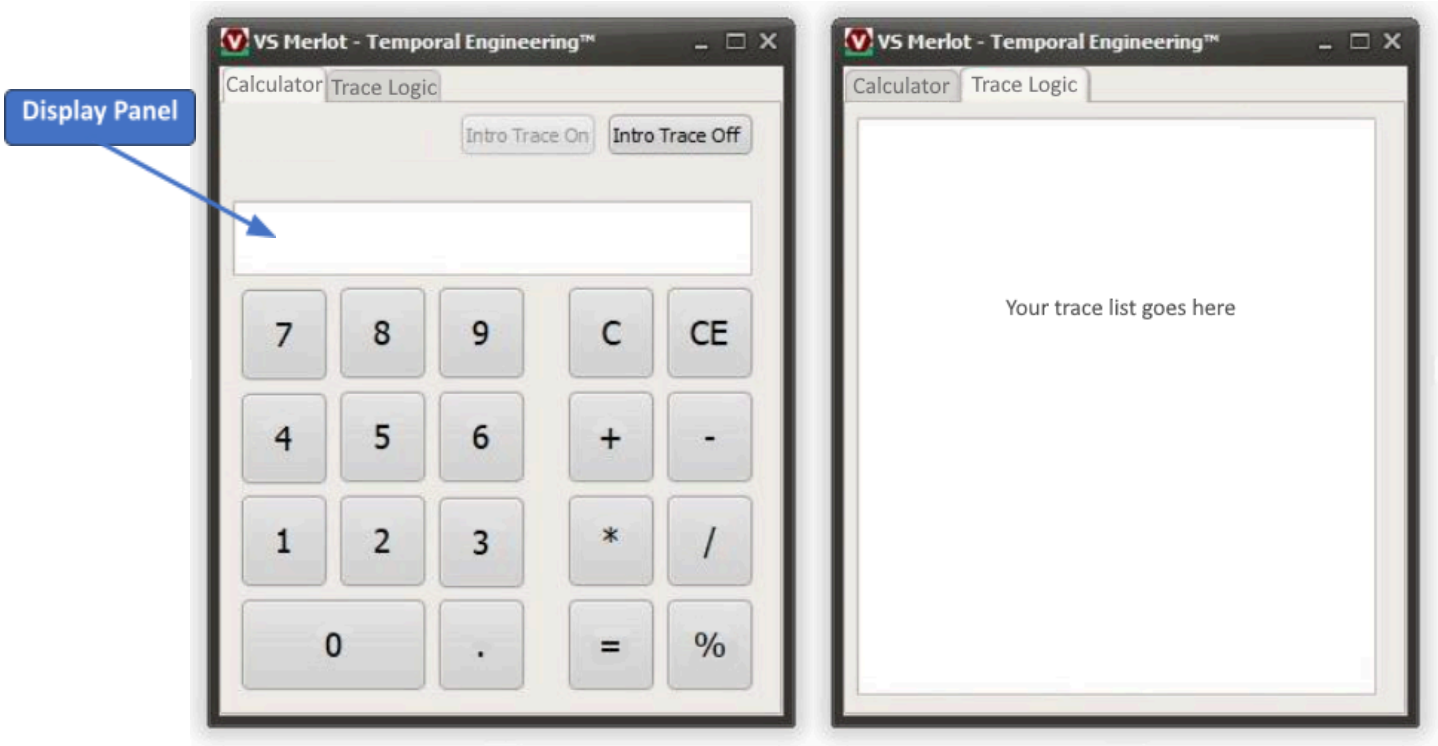
This is the specifications and requirements document for developing a five-function Calculator. This assignment helps us evaluate your approach to software development and is useful as a framework for further conversations regarding Temporal Engineering.

When complete email your calculator to Tom Bowman (tom.bowman@vsmerlot.com).

Please include project files and a readme file with any relevant notes or comments. Also, be ready to demo your calculator for follow-up interviews.

The User Interface

Below is an example of the calculator user interface design. Design your calculator to follow this general layout and design of this example, but it does not need to be an exact copy. The buttons and functionality for the “Calculator” and “Trace Logic” tabs are explained on page 2.



Calculator Tab:

Graphical Components:

- **Button Components:**
  - **Numbers 0-9**
  - **Decimal**

- **Operators (+, -, *, /, %)**
    - Addition, Subtraction, Multiplication, and Division are all standard calculations.
    - **Subtraction/Negative button:** Use the subtraction button for both: a subtraction operation and a negative sign.
    - **Percentage button:** This calculation is an additive tax calculation. For example, adding a five percent tax to one hundred dollars (100 + 5%) equals 105. After selecting button *5*, select the percentage button to perform the additive percentage operation.
  - **C button:** Clear the entire calculator's state and display
  - **CE button:** Clear the current entry
  - **Intro Trace On button:** Turns the introspective trace on
  - **Intro Trace Off button:** Turns the introspective trace off
- **Display Components:**
  - **Display Panel**: Display's the most recently selected number and the result after "=" is selected.

Trace Logic Tab:

Every decision point must be logged and displayed in a list format in the Trace tab (this includes *every* relevant branching decision point, i.e. *if*, *else*, *switch*, etc.). The trace should show the latest input, the current calculator state, the current decision point being reached, and provide a running count of decisions required to process the current input.

Technical Requirements:

To ensure reproducibility, please follow these requirements so that your calculator may be built and run by VS Merlot:

- Use Microsoft Visual Studio 2022 or newer (Community Edition) and the MSVC C++ Compiler.
- Must use the standard C++ language implementation, not the Managed C++ language created by Microsoft (sometimes referred to as C++/CLI).
- The calculator application must target and function correctly on Windows (10 or newer).
- User interfaces must only use the provided Windows graphics libraries and components bundled with MSVC.
- Only use default MSVC-provided libraries, no external/third party libraries allowed.
- ZIP and email the entire Visual Studio project or send a link to a Github repository containing the project.

Contact information:

You can reach us via email at: tom.bowman@vsmerlot.com

You can reach us via phone at: 903-987-0168