**2022 Spring CmpE 230 Hw3**

**Zeynep Buse Aydın** 2019400066 – **Asude Ebrar Kızıloğlu** 2019400009

In this Qt project, to generate a standard hexadecimal calculator with only addition and subtraction functions, we mainly used a class ‘*calculate.h*’ && ‘calculate.cpp’ and a user interface file ‘*calculate.ui*’. Below, the content of the calculate.cpp (with the functions from ‘*calculate.h*’) is explained and the general information about the design is given. For the numerical buttons of the calculator, **NumberPressed**() function is processed. For the buttons ‘+’ and ‘-’, **MathButtonPressed**() function; for ‘=’, **EqualPressed**() function; and for the ‘clr’ button, **ClearPressed**() function is called.

**Variables in calculate.cpp:**

**long result**: stores the result of the operations at any time – cleared at the end of Equal and Clear functions.

**bool ok:** dumb var to use QString::toLong(&ok, 16) –hexadecimal

**Some flags:**

**bool addPressed / subPressed**: Respectively check whether ‘+’ or ‘-’ is just pressed.

**bool endOfNum**: Check whether a number is finished to be entered. This occurs whenever ‘+’ | ‘-’| ‘=’ is pressed.

**bool firstOperation**: At the beginning, or once ‘clr’ is pressed, first number is entered. This flag checks it.

**Functions in calculate.cpp:**

**Calculate::Calculate:**

In this main body, buttons from ui are connected to the relative functions.

* Number buttons are added to the QPushButton \*numButtons[16] array before being connected to **NumberPressed**() function, to facilitate this connection.
* Other buttons are connected to the corresponding functions.

**Calculate::NumberPressed:**

* If the previous number is ended (**endOfNum**) or current display is 0, the number pressed is displayed directly.
* Otherwise, the number pressed is concatenated to the string of the previous display number.

**Calculate::MathButtonPressed:**

* Current display is put into displayValue.
* If this is the first operation (flag check), displayValue is assigned to **result**.
* Else if either **addPressed** or **subPressed** is true, the corresponding operation is made onto **result** and display is updated.
* According to the current operator pressed, relative boolean flag is marked true.
* **endOfNum** is assigned to true.

**Calculate::EqualPressed:**

* If either **addPressed** or **subPressed** is true, the corresponding operation is made onto **result.**
* **result** is displayed.
* The flags (**addPressed**, **subPressed**, **firstOperation**, **endOfNum**) are marked as necessary, and **result** is set to 0.

**Calculate::ClearPressed:**

* Display is cleared.
* The flags (**addPressed**, **subPressed**, **firstOperation**, **endOfNum**) are marked as necessary, and **result** is set to 0.

**Design:**

We used the *‘calculate.ui’* file that is generated from the design that we did with drag-and-drop in QT Designer.

* There are in total 20 push buttons in the calculator:
  + 16 number buttons (from 0 to F),
  + 1 equals button,
  + 1 addition button,
  + 1 subtraction button,
  + 1 clear button.
* All the buttons are put in grid layout.
* The current result is displayed in the line edit widget.