

# CMPE 230 ASSIGNMENT 3 REPORT

Elif Çalışkan 2016400183

Yağmur Kahyaoğlu 2015400057

## HOW TO COMPILE:

Execute the program with a suitable QT IDE.

## IMPORTANT NOTES:

- Clear button makes the current label 0.
- User can switch between “hex” and “dec” modes at every situation.
- In “dec” mode, the letter buttons (A, B, C, D, E, F) that are special for “hex” mode, work like clear button.

## 1)PROBLEM DESCRIPTION

In this project we were asked to develop a QT program that implements a decimal/hexadecimal calculator. When “Hex” button is pressed, the calculator should operate in hexadecimal mode. When “Dec” is pressed, it should work in decimal mode.

## 2)PROBLEM SOLUTION

**MainWindow::MainWindow(QWidget \*parent) :**  
**QMainWindow(parent), ui(new Ui::MainWindow)**

We connected every button that is a digit with the slot digit\_pressed(). They get activated with signal released(). We connected every button that is an operator with the slot operator\_pressed(). They get activated with signal released(). We set the buttons checkable to check later.

**void MainWindow::digit\_pressed() :**

We use this function for displaying the digits that are pressed in an appropriate way first we check if we are in mode hex, if any operator is checked (which means the variable take is false) and the user is not typing the text on the button is displayed. Else if equals operator was called before pressing the button it also displays the text on button. Else, it appends the text on button to the label's text and displays the new string. If it is in dec mode, we do the similar things except we don't cast values to integer. Also, we don't append the new text if the current text is "0", we simply write the new button text.

**void MainWindow::calculate(int second, int op) :**

This function takes two parameters, second and op, and makes the calculating operations. If op is 2, it adds the second value to result, else if op is it subtracts second from result. Else if it is 4, it multiplies second with result and else if it is 5, if second is not 0 it divides result with second, else it displays an error message. In every case the function also checks the calculator's mode and if it is in hex mode, it converts the value of result to hexadecimal before displaying it.

### **void MainWindow::on\_pushButton\_equals\_released() :**

First we set equals to true. Then we check if the mode is hex. If it is, we assign the label text to hexnum. We call the toUInt function to convert hexnum to decimal and store it in second. Else, we only store the text on label as integer. We calculate the result by passing second and op as parameters. We assign zero to result since the previous calculations is done. We assign zero to op so that it is not confused with any other operator. We assign false to take so that it means no operator is checked at the moment. It is used in digit\_pressed().

### **void MainWindow::on\_pushButton\_clear\_released() :**

If clear is pressed it makes the label text "0".

### **void MainWindow::operator\_pressed():**

If an operator is pressed first we declare two integer variables: op1 and first. First is the first operand of any operator. op1 is the integer value for every operation that is currently checked. We check if the calculator is in hex mode. If it is, we convert the value on label to decimal and store it in first. Else we turn the label text to integer and change the value of checked of the button to true. op is the previous operation. If op equals 0, it means the calculator has just opened or the previous operator was equals. We change op1 value according to the checked operator. Then we change setChecked value of the pressed button to false. We assign false to take, because take variable is used for any operator that was currently checked. It is used for digit\_pressed() method. Then we calculate the result by passing second and op as parameters and op gets the value of op1.

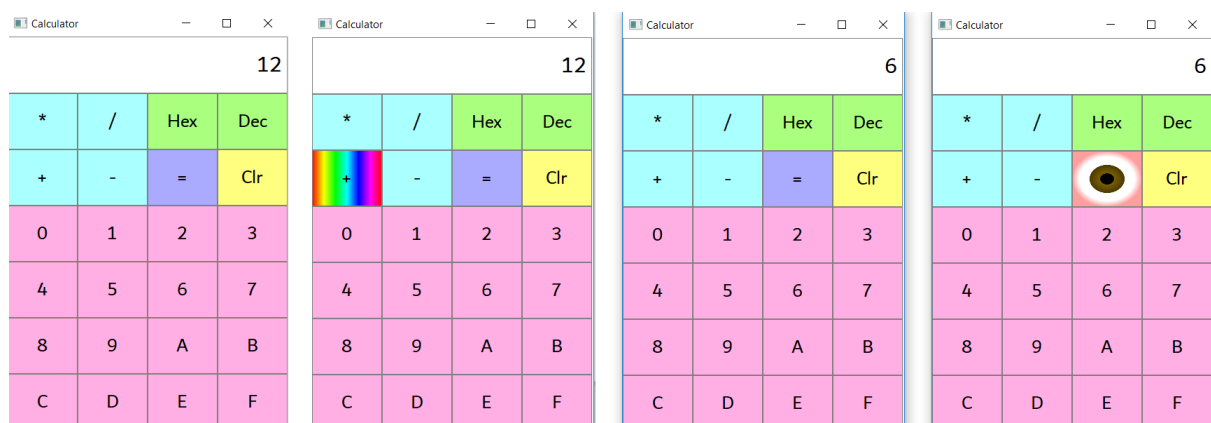
### **void MainWindow::on\_pushButton\_hex\_released() :**

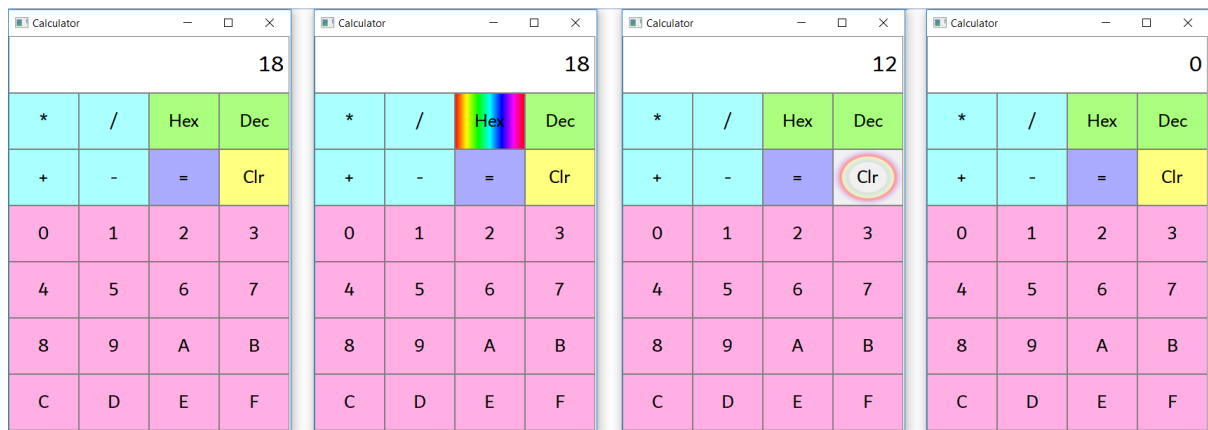
If hex button is pressed, if the calculator wasn't on the hex mode already, we convert the current text on the label to hexadecimal and display it. We also make isHex true. After that, every operation is executed in hex mode until dec is pressed.

### **void MainWindow::on\_pushButton\_dec\_released() :**

If dec button is pressed, if the calculator wasn't on the dec mode already, we convert the current text on the label to decimal and display it. We also make isHex false. After that, every operation is executed in dec mode until hex is pressed.

## **3)EXAMPLES**





#### 4)CONCLUSION

To sum up, we implemented a QT program that implements a decimal/hexadecimal calculator. When “Hex” button is pressed, the calculator operates in hexadecimal mode. When “Dec” is pressed, it works in decimal mode. We added buttons and a text label for the calculator. We designed the buttons and gave them functionalities according to their roles in calculator.