**CS 135 -- Lab 6**

**Introduction**

The purpose of this Lab is a warm-up for Project 2. It will familiarize you with the UI (user interface) specification, and teach how to use the random number generator, using *srand()*, and *rand()*.

**Instructions**

What we are doing for this week's coding projects:

1) Create a new folder: `Desktop > code_2020-10-19`
2) Use EditPlus to create three C++ Project as follows:

   Programming Challenges from the textbook. ***Be sure to read the instructions!***

| | | Project Name | Source Code |
|---|---|---|---|
| Chap 3: p. 147 | 17. | **Math_Tutor** | *Math_Tutor.cpp* |
| Chap 4: p. 224 | 11. | **Math_Tutor2** | *Math_Tutor2.cpp* |
| Chap 5: p. 298 | 8. | **Math_Tutor3** | *Math_Tutor3.cpp* |

The idea is to work through the 3 projects progressively, and add the specified features.
**Math_Tutor3** is the final product that you will be submitting.

## Sample Outputs

Sample output for *Math_Tutor3*:

```
     Math Tutor Menu
------------------------------
1. Addition problem
2. Subtraction problem
3. Multiplication problem
4. Division problem
5. Quit this program
------------------------------
Enter your choice (1-5): 3


    31
 *   9
  ----
   279


Congratulations! That's right.

     Math Tutor Menu
------------------------------
1. Addition problem
2. Subtraction problem
3. Multiplication problem
4. Division problem
5. Quit this program
------------------------------
Enter your choice (1-5): 4


  196 / 7 = 28


Congratulations! That's right.
```

**Discussion**

How the lessons learned from *Math_Tutor3* apply to *Project 2*.

- Both projects are at the 5th Grade math level.
- Both projects have the same presentation layout.
- *Math_Tutor3*: Add and Subtract only up to 3-digit numbers.
- *Project 2*: Add and Subtract from 1 to 5-digit numbers, as chosen by the user.
- Subtraction: designed so there are no negative answers
- Multiplication: answers will have about twice as many digits as the operands.
- Division: designed to be even, without remainder.
- *Math_Tutor3*: has only one correct answer.
- *Project 2*: has multiple choice answers. These should be designed so they are not *too* obvious.

**Coding Convention**

- Each file shall have a file header.
- Each function shall have a function header.
- Your code shall be properly indented and commented.
- Your code shall include the lines in the *pgm_template.cpp* at the end of the main() function that will output your name and the date, and pause the program.

**Submission Instructions**

Compile and test your code in the MinGW environment we have on campus. That is how it will be graded.
The latest version should be compiled and running, and that is what will be tested for grading.
The executable shall be able to run correctly by double-clicking on it.

The `code_2020-10-19` folder should contain the the following folders and contents:

- **Math_Tutor3**, containing:
  - source code: *Math_Tutor3.cpp*
  - executable: *Math_Tutor3.exe*

Do not submit the earlier projects, only **Math_Tutor3**.

Zip up your folder and submit your ***code_2020-10-19.zip*** file to the Canvas Drop box.