CSCI-1200 Data Structures — Fall 2025
Homework 1 — Math Equation Game

This homework is an extension and enhancement of **Homework 0 Guessing Game**. The purpose of the assignment is to continue practicing with C++ development, add error checking, support an additional feature (equation mode), and refactor and improve the structure and readability of your program in preparation for TA manual grading of the code.

IMPORTANT NOTE: If you did not complete Homework 0 and receive most or all of the autograding points on Submitty for Homework 0, please continue to work on and debug your implementation before proceeding to the additional features needed for this new homework.

The example on the right demonstrates the error checking you should add to your program from the previous assignment. Helpful error messages should be printed to `std::cerr` when the user has made a mistake in the command line arguments or in making a guess. The convention in C++ is to print normal, non-error program output to `std::cout` and error messages to `std::cerr`. With typical interactive play at the keyboard, both of these streams are displayed interspersed to the terminal. But when we test your program on Submitty, we will separate these two streams into different files.

Study the provided examples closely. If a specific guess includes multiple errors, your program can print a message about any or all of the errors. That is, we will not grade strictly on the error message for a guess with multiple mistakes (e.g., if the input is both the wrong length *and* it is not a valid equation).

The primary new game feature for this homework is that the guesses will be mathematical equations. In addition to the digits '0' - '9', the input will have one or more operators ('+', '-', '*', and '/') and an equal sign '='. If the guess is not a valid mathematical equation, your program should print an error message.

Similar to the previous assignment, for each (non-error) guess, the program will print an ASCII art visualization of the correctness of each character of the guess. A frame of '*'s indicate the character is in the correct position and a frame of '.'s indicate the character is in the secret, but in a different position.

```
> ./game.out
ERROR: missing filename on the command line
> ./game.out does_not_exist.txt
ERROR: Could not open 'does_not_exist.txt' for reading
> ./game.out sample_equation.txt
Let's play a guessing game!
The secret is an equation with 7 characters.
your guess: 1+1=2
ERROR: wrong length
your guess: 1+1+1=2
ERROR: invalid equation
your guess: 1+1+1=3

+----------------------------------------+
|..... .....                 ..... .....|
|.   . .   .                 .   . .   .|
|. 1 . . + .   1     +     1   . = . . 3 .|
|.   . .   .                 .   . .   .|
|..... .....                 ..... .....|
+----------------------------------------+

your guess: 68+3=71

+----------------------------------------+
|           ***** ..... ***** ..... *****|
|           *   * .   . *   * .   . *   *|
|  6    8   * + * . 3 . * = * . 7 . * 1 *|
|           *   * .   . *   * .   . *   *|
|           ***** ..... ***** ..... *****|
+----------------------------------------+

your guess: 24+7=31

+----------------------------------------+
|***** ***** ***** ***** ***** ***** *****|
|*   * *   * *   * *   * *   * *   * *   *|
|* 2 * * 4 * * + * * 7 * * = * * 3 * * 1 *|
|*   * *   * *   * *   * *   * *   * *   *|
|***** ***** ***** ***** ***** ***** *****|
+----------------------------------------+

CONGRATULATIONS!
You solved the puzzle after making 3 guesses!
```

Suggestion to Get Started

We suggest you begin with 1-2 digit positive numbers and the addition operation. Getting this working will earn you the majority of the autograding points. When that is done, if you have time, you can add in the other operators, parse equations with negative numbers, handle the order of operations, and worry about non-integer values that may result from the division operation. But these final pieces will be worth a minor portion of the assignment grade.

Refactoring your Homework 0 Code

An important part of this assignment is the opportunity to revisit and improve the program that you submitted for Homework 0. The TAs will be manually grading your assignment for overall code quality. See "Good Programming Practices". This includes organizing your code into functions, eliminating redundant/repetitive code, adding and/or revising comments, making efficient use of data structures, and proper use of `const` and pass-by-reference, etc. Document in your README.txt file the details of one or two key changes you made to improve your code quality from your final Homework 0 submission.

Extra Credit: Challenge Mode

For a small amount of extra credit, you may add an additional *optional* argument to your program, '`--challenge_mode`', which will print additional types of error messages if the user makes a guess that is impossible given the previous guesses and correctness visualization. An example of this extra credit behavior is shown on the right.

Additional Notes

You should use the STL `string` and `vector` classes and practice correct use of `const` and *pass-by-reference/return-by-reference* when moving (large) objects between functions. You are not expected to use classes on this homework. Do not use data structures or concepts that we have not yet covered in lecture/lab.

```
> ./game.out sample_equation2.txt --challenge_mode
Let's play a guessing game!
The secret is an equation with 6 characters.
your guess: 11-8=3

+-----------------------------------+
|.....                   .....      |
|.   .                   .   .      |
|. 1 .   1     -     8   . = .   3  |
|.   .                   .   .      |
|.....                   .....      |
+-----------------------------------+

your guess: 7*3=21
ERROR: character '3' appears at most 0 time(s)
your guess: 6*4=24
ERROR: character '1' appears at least 1 time(s)
your guess: 7*2=14

+-----------------------------------+
|                  ***** ***** *****|
|                  *   * *   * *   *|
|  7     *     2   * = * * 1 * * 4 *|
|                  *   * *   * *   *|
|                  ***** ***** *****|
+-----------------------------------+

your guess: 4+9=13
ERROR: character [5] of the guess should be '4'
ERROR: character [5] of the guess cannot be '3'
ERROR: character '3' appears at most 0 time(s)
your guess: 5+9=14

+-----------------------------------+
|..... ***** ..... ***** ***** *****|
|.   . *   * .   . *   * *   * *   *|
|. 5 . * + * . 9 . * = * * 1 * * 4 *|
|.   . *   * .   . *   * *   * *   *|
|..... ***** ..... ***** ***** *****|
+-----------------------------------+

your guess: 9+5=14

+-----------------------------------+
|***** ***** ***** ***** ***** *****|
|*   * *   * *   * *   * *   * *   *|
|* 9 * * + * * 5 * * = * * 1 * * 4 *|
|*   * *   * *   * *   * *   * *   *|
|***** ***** ***** ***** ***** *****|
+-----------------------------------+

CONGRATULATIONS!
You solved the puzzle after making 4 guesses!
```

You must do this assignment on your own, as described in the "Collaboration Policy & Academic Integrity" document. List the names of anyone you talked to about the problem or debugging and all references you consulted in preparing your solution. Please attend "TA and Mentor Office Hours". with your questions about the assignment and for help on your development and debugging process. Ask short questions about the homework and lecture material on the "Discussion Forum".