

For homework #3 you are to write an improved Math Tutor program for small children. Your program's interaction should look something like: (Student input shown in **bold**.)

```

Welcome to the Math Tutor

Enter your name: Bob<Enter>
Enter your grade (1, 2, or 3): First<Enter>
Sorry, your choices are the numbers 1, 2, or 3.
Enter your grade (1, 2, or 3): 3<Enter>

Please answer the following 10 problems Bob.

Problem 1:  395
           + 137
           ----
Answer:     422<Enter>

Sorry Bob, but your answer is incorrect.  The correct answer is 532.
.
.
.
Problem 10:  728
            + 123
            ----
Answer:     851<Enter>

Great Bob!  You are correct.

=====
Summary:  Bob you got 7 correct and 3 incorrect, so you got 70.0% correct!

```

Your program should use the following table based on the grade level to select the random numbers for the problems **and number** of the problems.

Grade Level	Range of the Random Numbers for Problems	Number of Problems
1 (first grade)	0 to 9	5
2 (second grade)	10 to 99	7
3 (third grade)	100 to 999	10

Your program also needs to:

- alignment the numbers to be added,
- check for a valid grade selection. Anything other than 1, 2, or 3 should **repeatedly ask them to re-enter a choice until it is valid**.
- track the number of correct answers so a summary can be printed at the end that includes # correct, # incorrect, and their percentage correct

Extra credit feature:

- Append the summary information for each execution of the program to the file: `mathTutorLog.txt`, so the teacher can review students' progress. Append a line containing name, grade, # correct, # incorrect, and percentage correct. To append to a text file, you'll need to create an `fstream` object and set its access flag to append mode by (see pp. 652-3 of textbook for more detail):

```

fstream logFile;
logFile.open("mathTutorLog.txt", ios::out | ios::app);
logFile << name;
....
logFile.close();

```

When you write your program, be sure you:

- use meaningful variable names with good style (i.e., useCamelCase)
- use comments (`//` single-line or `/*` Multi-line Comment `*/`) at the start of the program, to label tricky blocks of code, and to explain the contents of variables
- use global constants where appropriate with good style (`ALL_CAPS_AND_UNDERSCORES`). (Put your global constants after your initial comments describing the program and before your main function definition so they can be found and changed easily in future versions of your program.)

Submit the single file, hw3.zip containing the following:

- **mathTutorHW3.cpp** (your C++ program)
- **out.txt** script file demonstrating the features of your program as it runs (recall the UNIX/Linux command “script out.txt” starts capturing everything that goes across the screen, then run the program, and finally Ctrl-d stops the script)
- **mathTutorLog.txt** file if you did the extra credit part

The steps for the homework submission system are:

1. Design, write, debug, and test your program in a new hw3 subdirectory on student.cs.uni.edu. When you are ready to submit your homework, copy it back to your computer using FileZilla (or WinSCP) program, zip the whole hw3 folder on your computer by right-clicking on it and selecting Send to | Compressed (zipped) folder. This will create a new file called hw3.zip which you will submit electronically. (This assumes Windows OS....)
 2. Log on to the submission system at: https://www.cs.uni.edu/~schafer/submit/which_course.cgi
(It is very likely that you will get some security certificate warnings when trying to use this. You may add an exception and accept the existing security certificate.) Use the same CatID user-name and password you use to log on the lab computers.
 3. Select the course and section number of "CS 1160, C/C++ Programming, Fienup". Click the "Continue".
 4. Select the homework that you wish to submit: "HW 3: Math Tutor Improved". Click the "Continue" button.
 5. Specify how many extra files you want to submit. Just leave it at 0. Click the "Continue" button.
 6. Upload your program by Browsing and selecting your hw3.zip file. Click the "Continue" button.
 7. The next page reports on the status of the upload(s). You can always continue to upload a better version of the program until the deadline. The newer file will replace an older file of the same name.
- (If you miss the deadline, you'll need to submit it as above, but select “Late Homeworks” in step 4 above.)