CmpE 150 Introduction to Computing, Fall 2015
Project 1—Due:28/10/2015, 11am

You will write a Java program to teach simple math with two numbers n and m. Since you are the teacher, the text that you produce will have some gaps for the kids to fill in.

- Write the mupliplication table such that the table goes up to $n*n$ but leave the symmetric positions empty for the kids to figure out. For example, if n is 4, your table should look like this:

  ```
  1
  2 4
  3 6 9
  4 8 12 16
  ```

- Write a series such that each number finds the number of digits in the previous number and multiplies that with n. Since for the first number, there is no previous number m is taken as the previous number. Continue this for 4 iterations. For example, if n is 4 and m is 23415, the series will produce the following:

  ```
  20
  8
  4
  4
  ```

Program:

1. Your program should have at least two static methods in addition to your main method. Each method corresponding to one of the two analysis above.
2. You should have at least two constants, which hold the two numbers n and m.
3. You are not allowed to compute the above analysis by hand and then use println statements to print the result. All computations must be done by your program.
4. You are not allowed to use statements that we haven't learned in class yet (such as if statements, while loops, and so on).

Submission: You will submit a project report and your code over Moodle. Your program should be named with your initials and your student number together (e.g., OS2013800027).

Project report should consist of five sections. These are:

1. Problem Description: In this section, you should describe the problem in your words.
2. Problem Solution: In this section, you should specify the concepts (methods, for loop, etc.) that you use in your program. Explain each one (i.e. why you need it, what you accomplish by using it, so on.). Report how many for loops you use.

3. Implementation: This section will include your whole code with comments. You need to pay attention to indentation in order to improve readability.
   - Do not forget to explain each variable that you use (i.e. int count=0; // count is the number of items).
   - Before each method, specify what the method does (i.e. /* This method . */)
4. Output of the program: A screen-shot of your program output should be put in this section.
5. Conclusion: You should evaluate your work here. State whether you have solved the problem correctly. If not, state what is missing, what could have been improved, and so on.

You will submit these over Moodle as a single zip file where the file name is your student number. Your zip file should consist of your .java file and your report in .doc or .pdf format.

Partial Submission: If you cannot do both of the analysis above, you should still submit your code as well as your report. Try to do as much as possible.

Late Submission: Any submission after the deadline is considered late and will not be accepted.