**CSE 174 – Fall 2016**

**PROGRAM #6: 40 points – Due Sunday, October 9, by 11:59 p.m.**

**Outcomes:**

* Write programs that obtain user input
* Write programs that use if and else statements
* Write programs that work with random numbers
* Write programs that display numbers formatted according to a given specification
* Format and comment source code that adheres to a given set of formatting guidelines

**Scoring:**

At a bare minimum, the program you submit must have the assigned source code, and your source code must compile and run without crashing.

* If you do not submit a zip file containing your source code (.java file), your score will be zero.
* If you submit source code that does not compile, your score will be zero.
* If you submit source code that roughly resembles the requirements and it compiles, but it crashes under normal operating conditions (nice input from the user), your score will be reduced by 75%.
* Deductions will be made for not meeting the usual requirements:
* Source code that is not formatted according to guidelines
* File and class names that do not meet specifications

|  |  |  |
| --- | --- | --- |
|  | **Full credit** | **No credit or Partial credit** |
| **Generate random problems**  **(12 points)** | You used Math.random() to generate random multiplication and division problems as specified. Your random numbers allow for all possible multiplication problems up to the limit set by the user. | Numbers are not random, or do not conform to specifications. |
| **Check user answers**  **(12 points)** | You determine whether the user correctly solves the 6 problems. | There are errors in determining whether the user answer is correct. |
| **Display statistical results**  **(12 points)** | You correctly compute all the numerical results, including ongoing count, the elapsed time, and the final counts and percentages. | There are errors in your computations, or some computations are missing. |
| **Format console output**  **(4 points)** | You formatted output as specified, including rounding percentages to the correct number of decimal places. | Screen output does not match specifications. |

**Preliminaries:**

* Review Java's Math.random() method. Although this method produces double values from 0 to almost 1, it is possible to get almost any other range of values you want by using some combination of addition, multiplication, and possibly casting as an int. For example, the following will produce random integer values from 1 through 6, inclusive:

(int)(1 + 6 \* Math.random())

This produces random integer values from 20 through 23, inclusive:

(int)(20 + 4 \* Math.random())

* Look online to find a way to compute elapsed time in Java. The timer should start when the user sees the first question, and end when the user answers the last question.

**Assignment:**

Write a program that gives the user 3 random multiplication problems, followed by 3 random division problems to solve. Display the user's ongoing progress, final scores, and elapsed time. Your program should allow the user to select the difficulty level of the problems, and should match the formatting shown in the sample run as closely as possible.

**Requirements:**

Your program should meet all of the following requirements:

* The class name should be MathGame
* All 6 problems should use int values, including the answers.
* Use Math.random() for generating your random values.
* The range of random numbers for each problem should be limited based on user input.
  + In multiplication, the limit the size of the answer to the multiplication problem. For example, setting a multiplication limit of 100 means the answer to each multiplication problem will range from 1 to 100 inclusive.
  + In division, the limit set by the user will limit the size of the first number in the division problem. For example, setting a division limit of 100 means that the first number in the division problem will range from 1 to 100 inclusive.
* For multiplication, generate your random numbers so that it is possible to get any of the possible multiplication problems up to the limit. For example, if the user sets a multiplication limit of 6, then all of these problems should be possible: 1 \* 6, 2 \* 3, 3 \* 2, and 6 \* 1.
* For division problems, there should never be a remainder. So, 48 divided by 4 should be possible, but 48 divided by 5 should not be possible.
* Display the ongoing scores and final scores exactly as shown above, including rounding percentages to three places after the decimal point.
* Display the total time it took the user to answer all 6 questions.
* Put your MathGame.java file in a folder named program6, with no other files. Zip the program6 folder and submit that zip file. The name of the zip file does not matter.

**Sample run:**

**Your program should match this format as closely as possible. Note that text shown in red is there because the user typed it. You are not supposed to print those.**

|  |
| --- |
| Welcome to my math quiz!  This quiz will give you three multiplication problems,  and then three division problems.  -----------------------------------------------------  Enter the multiplication limit: **100**  Enter the division limit: **50**  The timer starts...now!  -MULTIPLICATION--------------------------------------  5 \* 10 = **50**  Yes! You have 1 correct so far.  8 \* 7 = **56**  Yes! You have 2 correct so far.  8 \* 9 = **71**  Sorry, 8 \* 9 = 72. You have 2 correct so far.  -DIVISION--------------------------------------------  42 / 6 = **5**  Sorry, 42 / 6 = 7. You have 2 correct so far.  50 / 5 = **10**  Yes! You have 3 correct so far.  1 / 1 = **0**  Sorry, 1 / 1 = 1. You have 3 correct so far.  The timer stops...now! You answered in 47 seconds.  -RESULTS---------------------------------------------  Multiplication score: 2 out of 3 (66.667%)  Division score: 1 out of 3 (33.333%)  Overall score: 3 out of 6 (50.000%) |