**The Math Program.**

* **Level one: [EASY LEVEL]**
  + Here on level one, we create 5 random numbers, and all these random numbers must be less than 10.
  + Then we create an int variable and we add all these 5 random numbers.
  + Then we display these 5 random numbers to the player and ask the user for the answer(Do this for all the stages in this level) (meaning we create another int variable that is going to store the answer of the user), then if the answer given by the user is the same as the answer generated by the computer( the user goes to the next stage (2) (note: the user can not just go back to the previous stage, he/she must finish all the stages )[ this is addition]
  + We continue to do this for Subtraction, Multiplication, Division and Modulus.
  + The first level will have 10 stages, meaning that for each Mathematical operator, there will be 2 stages.
  + There will be 3 hints and 3 tries, for each hint used we subtract 1 from the number of tries.(This is optional: we will try to make the user to get more hints if his/her hints are less than 3, but the catch here is that in order for the user to get a hint , he/she must get 2 alternating stages correctly in the first try).
  + **THE CODE:**
    - First create a function.
    - In this function create 5 random Integers, and these integers must contain numbers that are less than 10.
    - After creating these random numbers, create another three integers (2 of those integers must have the value 0 and the third one a value 3)
    - In Those two integers that have 0 on them , one of them must be for allowing the user to repeat his/her answer 3 times if he/she doesn’t get the answer right, and the second one must be for allowing the user to repeat his/her input 3 time if she/he made a mistake by assigning his/her answer as a string instead of an integer, the integer with the value 3, is just for tries.
    - Then create a Scanner that will take user input.
    - Create a loop that will compare user input with the answer that we want.
    - Inside the loop create a try and catch method that will give the user another chance if he/she didn’t get the answer in the first try.
    - Repeat this code for the other 10 remaining stages.
    - Example of the mathematical equation that will be displayed for the user: 1+3+2+9+1=
* **LEVEL TWO: [NORMAL LEVEL]**
  + Here we also still show 5 numbers which also created randomly.
  + The catch here is that instead of using 1 mathematical operator we use two.
  + These created random numbers must be 10<= x <=100.
  + **THIS IS HOW THE MATHEMATICAL OPERATORS WILL BE FROM STAGE 1 TO STAGE 9:**
    - Here we are forcing the user to use BODMAS.
    - Addition & Subtraction
    - Multiplication & Division.
    - Multiplication & Addition
    - Division & Modulus
    - Multiplication & Subtraction
    - Division & Addition
    - Modulus & Addition
    - Modulus & Subtraction
    - Modulus and Multiplication
    - Use the same **code** as the one on level one but just do some lil bit of tweaking on it.
    - Each mathematical operation will have 1 stage here.
    - Example of the mathematical equation that will be displayed for the user: (1+3) +(2-9-1) =
* **LEVEL THREE: [HARD LEVEL]**
  + Here we also still have those five random created numbers
  + The catch here is that we use 3 or more mathematical operations
  + Here we include all numbers from zero to infinity, there’s no restriction
  + **THIS IS HOW THE MATHEMATICAL OPERATORS WILL FOLLOW EACH OTHER:**
    - Multiplication + Division + Random
    - Division – Addition + Random number
    - Division – (Modulus \* Random number)
    - Random number + Modulus + Modulus
    - Multiplication + (multiplication \* Random number)
    - Addition + addition + Random number
    - Addition – multiplication + Random number
    - Multiplication – Subtraction – Random number
    - Modulus +Multiplication + Random number
    - Subtraction - Subtraction – Subtraction
    - Create a bonus question: Question 11 (This is the longest question)
      * [multiplication \* subtraction] - [division \* modulus] + [addition \* multiplication] / [subtraction \* division] % [ addition \* subtraction] \* [ multiplication \* division].

**THE END**