Quarter 2 Programming Test Practice: The Calculator Project

Write a Java class that will accept *two integers (**A** and **B**) as input*, and then perform the following operations:

□	1. Add A and B
□	2. Subtract B from B
□	3. Multiply A and B
□	4. Divide A and B
□	5. Calculate A modulo B
□	6. Check if B is a divisor of A
□	7. Calculate $\log_B(A)$ –you may use *Math.log(x)* to help you calculate this (log A/log B)
□	8. Calculate A to the power of B (don't use Math.pow(x), B may be ±)
□	9. Check if A and B are primes or not
□	10. Output the result properly formatted, with commas separating the thousands

Your main method should only input the two numbers, and then call all 10 methods (functions) as appropriately. You may want to add a switch/case structure to implement a menu if you want some extra practice.

Using the code that we have written in class during this semester, re-write the algorithms into methods (functions) so that they can be easily debugged, called and reused.

Quarter 2 Programming Test Practice: The Calculator Project

Write a Java class that will accept *two integers (**A** and **B**) as input*, and then perform the following operations:

□	1. Add A and B
□	2. Subtract B from B
□	3. Multiply A and B
□	4. Divide A and B
□	5. Calculate A modulo B
□	6. Check if B is a divisor of A
□	7. Calculate $\log_B(A)$ –you may use *Math.log(x)* to help you calculate this (log A/log B)
□	8. Calculate A to the power of B (don't use Math.pow(x), B may be ±)
□	9. Check if A and B are primes or not
□	10. Output the result properly formatted, with commas separating the thousands

Your main method should only input the two numbers, and then call all 10 methods (functions) as appropriately. You may want to add a switch/case structure to implement a menu if you want some extra practice.

Using the code that we have written in class during this semester, re-write the algorithms into methods (functions) so that they can be easily debugged, called and reused.