**General Requirements**

Please review the following calculator programs and determine what they do in detail:

* + [https://www.calculator.net/binary-calculator.html (Links to an external site.)](https://www.calculator.net/binary-calculator.html)
  + [https://www.calculator.net/hex-calculator.html (Links to an external site.)](https://www.calculator.net/hex-calculator.html)
  + [https://www.calculator.net/bandwidth-calculator.html (Links to an external site.)](https://www.calculator.net/bandwidth-calculator.html)

If you don't know about binary and hexadecimal numbers, some information is provided on them and how the above calculations/conversions are performed on the same web pages.

1. List all the functions provided by each calculator. You should write a single program that supports the full functionality of the above three calculators.
2. Your program should have a user-friendly command-line interface (CLI) that allows the user to select the calculator type, a computation/conversion function, and enter the input values. It should do the computation or conversion and display the result to the user.
3. Your program should validate the input values entered by the user and prevent the entry of invalid inputs that will produce errors in calculations/conversions.

Design Constraints

1. You can have only one user-defined class in this assignment, and you can use only static methods in your implementation. This will be equivalent to doing “procedural programming” in Java.
2. You cannot use the available Java library methods to do calculations and conversions directly such as Integer.toBinaryString that will trivialize your work. This doesn't mean that you cannot use any other libraries of Java, e.g., String, Scanner, Math, Data Wrappers, etc.

Testing

1. Your program should provide a “test mode” for each type of calculator where it automatically runs all types of computations and conversions with some test input values and prints out the expected and actual results of them, and an error message if they don’t match. We call each of them a “test case.”
2. You can set the expected values for your test cases manually or use the Java library methods to do the computation/conversion on the fly to compute it.

Implementation

1. You should use IntelliJ IDEA for your implementation.
2. You should well document your code by using all three types of commenting effectively.