Programming Assignment #6: Arrays

Write: Implement the ArrayMethod application containing an array that stores eight integers. The application should call the following five methods:

1. display - should display all the integers
2. displayReverse - should display all the integers in reverse order
3. displaySum - should display the sum of the integers
4. displayLessThan - should display all values less than a limiting argument
5. displayHigherThanAverage - should display all values that are higher than the calculated average value.

Documentation: You must use a readable, logical, and coherent set of style and formatting rules. You are to stick to the "structured approach" in programming**.** Be sure to comment your code in addition to the required header. Each submission must have a block comment area that includes: Your first and last name, program exercise title, program due date, and the program description. You may also wish to comment your methods so that it is clear the parameters, purpose, and return values expected, if any.

Submission Details: All submissions are electronic. When you turn in a programming assignment, you must send me a compilable and correctly working copy of the assigned program source code. I will, at my discretion, compile and run (on my own test input) the programs you submit electronically. This is a part of my grading procedure. Your program must work. That means it must compile correctly, run according to specifications, and give correct results. Generally, a program that works will receive at least 40-50 percent of full credit. The rubric used for scoring is visible to you so please review it before you submit your assignment. Submit your source file for this program this means you are giving me your **<lastnameArrayMethod.java>** file not a link to an online compiler, text file, or executable file. In addition to your java file, include a word document that has screen snips for your test runs name that file **<lastnameArrayMethodTests.docx>.** *Your testing log needs to test the boundaries of your program. You should test your incorrect input statements work as designed and you should test each structure in your program. That should include being sure there are no infinite loops in your code or computations with negative numbers/letters etc.*

For full credit, your program must also meet the following criteria:

- Good design, including good algorithms.
- Good form, including documentation, and readability.
- Adequate testing, especially the testing of data boundaries and special cases.

You need to do a good job on all the criteria to receive an "A" on your program.