

ArrayList Manipulator of Integers Peer Review

Name _____

Date _____

Period _____

Peer Reviewer _____

**Peer reviewer initials below when
the ArrayList Manipulator of
Integer project is correctly
finished.**

_____ Style checked

_____ Code tested

Place a check mark (✓) when done correctly.

____ **Style and general code checked.**

- Comments the top with the student's name.
- Has correct default constructor.
- Passes CheckStyle.

____ **Code tested.**

- Option 1 creates random numbers between -500 and 500.
 - There is a random number of random numbers.
- Option 2 prints out the values in the array
 - Test with random data.
 - Test with Option 13 empty ArrayList. The code should not crash.
- Option 3 prints out sum
 - Test with random data.
 - Test with data from Option 12. The answer should be 329.
 - Test with Option 13 empty ArrayList. The answer should be 0.
- Option 4 prints out product
 - Test with random data.
 - Test with data from Option 12. The answer should be 5,167,968,750,000,000.
 - Test with Option 13 empty ArrayList. The answer should be 0.
- Option 5 prints out max and min.
 - Test with random data.
 - Test with data from Option 12. The answers should be Max = 50, Min = 7.
 - Test with Option 13 empty ArrayList. The answer should be Max = -2147483648, Min = 2147483647.

ArrayList Manipulator of Integers Peer Review

- Option 6 replaces all even numbers with 6.
 - Test with random data.
 - Test with data from Option 12. The answer should be {7,6,6,15,15,15,6,6,6,6,7}.
 - Test with Option 13 empty ArrayList. The code should not crash.
- Option 7 makes all numbers the absolute value.
 - Test with random data.
 - Test with Option 13 empty ArrayList. The code should not crash.
- Option 8 removes all odd numbers.
 - Test with random data.
 - Test with Option 13 empty ArrayList. The code should not crash.
- Option 9 duplicates each of the numbers that are a multiple of 3.
 - Test with random data.
 - Test with data from Option 12. The answer should be {7,10,10,15,15,15,15,15,15,50,50,50,50,50,7}.
 - Test with Option 13 empty ArrayList. The code should not crash.
- Option 10 removes one of each duplicate neighboring numbers.
 - Test with random data. (First select options 6 and 9 to test this.)
 - Test with data from Option 12. The answer should be {7,10,15,50,7}.
 - Test with Option 13 empty ArrayList. The code should not crash.
- Option 11 removes all neighboring duplicate numbers.
 - Test with random data. (First select options 6, 9, 9, 9, 9, 9 and 9 to test this.)
 - Test with data from Option 12. The answer should be an empty list.
 - Test with Option 13 empty ArrayList. The answer should be an empty list.
- Option 12 generates predictable data
- Option 13 generates empty ArrayList
- Option 14 says goodbye and exits