1. Test Plan
   1. Test for size of the array.
   2. Test for accuracy of the outputs.
   3. Check for proper execution of the requirements…ie: Constructor
   4. Test to make sure students are resetting values and just pulling values from old runs
2. Observations:
   1. NONE OF the versions passed all 3 of my tests.
      1. All of the students failed to set up a proper constructor.
      2. V1:
         1. Failed because the student got a -76 on my second test, which the student subtracted by -76 to get the correct answer based on the original documentation.
         2. Student failed to reset original value. Value would double every time the student ran the code
      3. V2:
         1. Failed on my 4th test, testing for size 5, expected 0 and received 2.
      4. V3:
         1. Failed my 3rd test because they hard coded the size for max size 3. I am testing for size 4, and it fails the test.
      5. V4:
         1. Failed my 3rd test, testing for array size 4.
      6. V5:
         1. Fails my test because it gives the wrong value for my 3rd test, should be value 5 and gives value 4.
3. Junit Test Suite:
   1. \*\*\*\*PLEASE NOTE: To test the different versions, I just change poly1 to whatever case I need (ie: poly 5 for V5).
4. **import** **static** org.junit.Assert.\*;
5. **import** org.junit.jupiter.api.BeforeAll;
6. **import** org.junit.jupiter.api.BeforeEach;
7. **import** org.junit.jupiter.api.Test;
8. **class** poly1Test {
10. poly2 p1 , p2, p3, p4;
12. @BeforeAll
13. **static** **void** setUpBeforeClass() **throws** Exception {
14. }
16. @BeforeEach
17. **void** setUp() **throws** Exception {
19. p1 = **new** poly2(3);
20. p1.setCoefficient(0, 3);
21. p1.setCoefficient(1, 5);
22. p1.setCoefficient(2, 0);
23. p1.setCoefficient(3, 2);
25. p2 = **new** poly2(2);
26. p2.setCoefficient(0, 0);
27. p2.setCoefficient(1, 0);
29. p3 = **new** poly2(4);
30. p3.setCoefficient(0, 1);
31. p3.setCoefficient(1, 1);
32. p3.setCoefficient(2, 1);
33. p3.setCoefficient(3, 1);
34. p3.setCoefficient(4, 1);
36. p4 = **new** poly2(5);
37. p4.setCoefficient(0, 2);
38. p4.setCoefficient(1, 2);
39. p4.setCoefficient(2, 2);
40. p4.setCoefficient(3, 2);
41. p4.setCoefficient(4, 2);
42. p4.setCoefficient(5, 2);
43. }
44. @Test
45. **void** evalTest() {
46. *assertEquals*("Error: eval test", 724, (**int**)p1.evaluate(7));
47. *assertEquals*("Error: eval test", 0, (**int**)p2.evaluate(0));
48. *assertEquals*("Error: eval test", 5, (**int**)p3.evaluate(1));
49. *assertEquals*("Error: eval test", 0, (**int**)p4.evaluate(0));
50. }
52. }