# Computer Science Contest #1920-14 Key

March 07, 2020

1) D
2) E
3) D
4) B
5) B
6) C
7) B
8) D
9) C
10) D

11) C
12) D
13) D
14) B
15) A
16) C
17) D
18) A
19) C
20) A

21) D
22) D
23) D
24) E
25) E
26) B
27) E
28) A
29) B
30) A

31) D
32) E
33) D
34) C
35) D
36) C
37) B
38) A
39) (0,1,1)
40) -31

**Note to Graders:**

- All provided code segments are intended to be syntactically correct, unless otherwise stated (e.g. error is an answer). **Ignore any typographical errors**.
- Any necessary Standard Java 2 Packages are assumed to have been imported as needed.
- Assume any undefined (undeclared) variables have been defined as used.

Brief Explanations:

1. DE - AD = 11011110 – 10101101 = 00110001
2. 5 + 7 % 9 * 9 / 4 – 3 = 5 + 7 * 9 / 4 – 3 = 5 + 63 / 4 – 3 = 5 + 15 – 3 = 20 – 3 = 17 => 17.0 when placed into a double.
3. %-5s means format 5 spaces, left justified, for the following String
4. Although x is not permanently changed, the method does return the altered String replacing any O with an A.
5. This is an Boolean algebra identity: A + !AB = A + B.
6. Math.floor returns a floating point value, and flooring a negative changes the value.
7. The compound operator += will automatically cast a double to an int.
8. Capitilization plays an important role in compareTo, a capital Y is considered to come before any lowercase letters.
9. Bob will print out at 0, 3, 6, thru 25. So 9 times.
10. list[7] is 5. list[5] is 3. Adding 1 makes it 4 and it is placed in list[6], which ironically already stores a 4.
11. To read from a file, you must call new File inside of an instantiated Scanner.
12. Multiplication by zero is zero. Ironically, this is a common error.
13. >> has a lower priority.
14. k – 52 unboxes the k into an int and creates another int. This resulting int cannot be autoboxed into a Double Wrapper class.
15. x.add adds to the end of the ArrayList. index numbering starts at 0.
16. 34 = 100010, 53 = 110010, the & will only turn on matching bits, so 100010 & 110010 = 100010 = 34.
17. (A && B || !(A || C)) && !B = (AB+!(A+C))!B = (AB+!A!C)!B =AB!B + !A!B!C= 0 + !A!B!C = !A!B!C, so (A, B, C) must be (F, F, F)
18. Don't forget that when you remove something from the ArrayList, the list shrinks and the next value could be skipped if you don't decrease the counter.
19. This is a ratio conversion, you start at m = 219 and need convert m to c by using m/6 and finding the remainder using m%6.
20. x/=10 returns the value x/10 and changes the value of x, so there is no error nor an infinite loop.
21&22. the val is not the asci character but rather the difference between that letter and 'A'.
23. A is an A and A is a C, but A is NOT a L, S, or Y
24. Since only S and Y can inherit the abstract class, you cannot have C inherit it. Because S and Y already inherit A, you cannot inherit a 2nd class. The better solution is to make T an interface.
25. Many new programmers confuse ArrayList with Array, Array cannot be resized and does not have a remove method.
26. Although x is consistently larger than the size of the array, x is being moded by the length of the array so that it is always inbounds.
27. x = 3, y = 3, so the base case goes off immediately
28. m(2,3) = 6 + m(3,3) + m(2,2) = 6, any value of by one is a multiple of both parameters.

m(2,4) = 8 + m(3,4) + m(2,3) = 8 + 12 + 6 = 26
m(1,4) = 4 + m(2,4) + m(1,3) = 4 + 26 + 11 = 41
m(1,3) = 3 + m(1,2) + m(2,3) = 3 + 2 + 6 = 11
The only value left is m(1,5) which must be 129.
29. A TreeSet is placed in comparable order when it entered into the set. The iterator removes the last value it passes over when a remove is called.
30. forEachRemaining is comparable to a for each loop, k is every object that the iterator hasn't gotten to yet, and the -> says what to do with the k.
31. When printing a String Set, the set is already in order of its comparable
32. addAll is a method in Set that returns a Boolean, not a Set.
33. The code adds 10,30 then removes 10 and adds 28,30,23,28,38 then removes 30,28 such that 30 is at the front of the queue.
34. MAX_VALUE = 0111 1111 1111 1111 1111 1111 1111 1111, when you shift over 29 bits you get 011 which is 3.
35. ""|B|""|A|N|A|N|A|""|A|""|L|""|A|""|B|""|A|M|A|""|A|T|L|""|A|NT|A
36. !(A+B)(!B(A+C))+!CB
    !A!B(!BA+!BC)+B!C
    !AA!B!B+!A!B!BC+B!C
    0+!A!BC+B!C
    !A!BC+B!C (this is a far as you can go, XOR does not occur because of !A)
37. LCIRC-2(201 XOR 221) OR RSHIFT-1(236)
    LCIRC-2(11001001 XOR 11011101) OR RSHIFT-2(11101100)
    LCIRC-2(00010100) OR 00111011
    01010000 OR 00111011 = 01111011 = 123
38. To do twos compliment, flip all the bits and add 1 to the results.
39. The digital circuit creates !ABC
40. - / + 21 19 10 * 5 7 = - / 40 10 35 = - 4 35 = -31