# Computer Science Contest #1213-08 Key

January 12, 2013

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

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

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

31) D
32) A
33) C
34) D
35) C
36) E
37) C
38) B
39) D
40) B

**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. 1010 ( 10 ) + 1010 ( 10 ) == 20
2. 1079 / 10 = 107
3. 1079 divided by 10 is 107 with a remainder of 9
4. The loop starts at 1 and increments by 4 stopping at 25. You get a run of 1 5 9 13 17 21 25
5. Substring(10) returns cirocks as c is at position 10 in the string
6. 3 + 4 is 7 as the sum of 3 + 4 is assigned to spot 0
7. The answer is 2 as 0 0 and 1 1 will make the statement true. ^ is the xor operator which means that only 1 can be true.
8. The sqrt of 40 is 6.32 which is bigger than 5.0. The output is 02 as the 2 is printed no matter the value of t.
9. E is the correct answer as B and C are correct and access the same variables in the same way.
10. The first case fails as 5 is too small and the second case works for both checks as 50 and 26 are big enough for both cases.
11. Tr is incremented to 6 before the check. The value 6 is assigned back to tr before the post incrementation. The post incrementation essentially does nothing in this situation.
12. The answer is (3.330) because the () signify that the number is negative.
13. The split array is [, big, , tall, , funny] as the split value is / as putting more than one / in the [ ] is the no different than putting just 1 /.
14. The answer is -4. You just have to walk through the letters in the string and make sure you see that the default substracts 1 for every letter not specified by a case.
15. lastIndexOf starts looking at length-1 and go from the end to spot 0.
16. 5 is the value assigned to the instance variable of ACat and the getIt() method returns the instance variable * 2.
17. The answer is 96 as the getIt() method from BigCat is called twice due to the fact that ACat is actually running from within BigCat.
18. 33 base 10 converted to base 16 is 21.
19. 2 in base 2 is 10 which is the value being accessed by the iterator
20. This question using bitwise operators. All bitwise operators use the binary value of each number. For 11 | 4, the code looks take 11 and converts it to binary – 1011. 1011 | 0100 is really what is going on. The result is 1111 as any column with a 1 results in a 1 in the answer. You need to study bitwise operators to do well on the uil written.
21. This question using bitwise operators. All bitwise operators use the binary value of each number. For 11 | 4, the code looks take 11 and converts it to binary – 1011. 1011 | 0100 is really what is going on. The result is 1111 as any column with a 1 results in a 1 in the answer. You need to study bitwise operators to do well on the uil written.
22. This question involves / and % using 11. % returns the remainder of division. You must know this for all UIL tests.
23. This is a pretty common parameter passing question involving references. All parameters in Java are passed by value meaning that a copy of the value is sent in. For this problem, a copy of the address of bob is passed to the method. This address can never be permanently changed. The code in the method attempts to point list at a new int[] array. This change only has local effect. None of the changes have any effect on bob.
24. This is a nested loop question with one loop that increments by 2 and one that increments by 1 but that starts iterating at the value of the

first loop variable. You must trace out the code and watch the values of the variables as they change. The runtime is n * n.

25. This is a nested loop question with one loop that increments by 2 and one that increments by 1 but that starts iterating at the value of the first loop variable. You must trace out the code and watch the values of the variables as they change. The runtime is n * n.
26. This is a nested loop question with one loop that increments by 2 and one that increments by 1 but that starts iterating at the value of the first loop variable. You must trace out the code and watch the values of the variables as they change. The runtime is n * n.
27. This is a matrix question that was patterned after a common UIL matrix question. Trace out the code and pay attention to where the variables are in the matrix.
28. This is a matrix question that was patterned after a common UIL matrix question. Trace out the code and pay attention to where the variables are in the matrix.
29. This question is a typical recursion question that only contains one call or branch. You must look for the patterns in the calls and start to recognize that calls with certain value always return the same value. This will allow you to cut down the amount of tracing. You can also use a program like JEliot to help you see the recursive calls.
30. This question is a typical recursion question that only contains one call or branch. You must look for the patterns in the calls and start to recognize that calls with certain value always return the same value. This will allow you to cut down the amount of tracing. You can also use a program like JEliot to help you see the recursive calls.
31. This code uses replaceAll and regex to replace all of the lowercase as with bs. + means 1 ore more and * means 0 or more in regex notation.
32. This question is using recursion to access the characters in a string. Most times, we use a loop to do this, but you could also use recursion and parameters. The code is summing up the ASCII values of the letters in the string.
33. This question is using recursion to access the characters in a string. Most times, we use a loop to do this, but you could also use recursion and parameters. The code is summing up the ASCII values of the letters in the string.
34. This question is showing how hierarchies work with inheritance. The child can never point to the parent, but the parent can always point at any of its children.
35. This is a simple array question using a for each loop and then using the actual value and mod to index this array. Typically, you do not use the for each loop value to index the array as it will cause an index out of bounds exception in most situations.
36. This question involves the use of a quick sort. Quick sort is pretty easy to recognize as it is recursive and it has a pivot typically without a temporary array ( like mergesort ). You need to spend some time tracing out the sorts and learning which sorts contain which code.
37. This question involves the use of a quick sort. Quick sort is pretty easy to recognize as it is recursive and it has a pivot typically without a temporary array ( like mergesort ). You need to spend some time tracing out the sorts and learning which sorts contain which code.
38. This is a typical loop tracing problem. J and sum are both changing as the loop iterates. Tracing this code on paper may be necessary.
39. You must trace out the tree on paper so you can see which nodes go where. It is very hard to try and work out a problem like this in your head although it can be done.

40. You must trace out the tree on paper so you can see which nodes go where. It is very hard to try and work out a problem like this in your head although it can be done.