Computer Science Contest #1314-09 Key

January 11, 2014

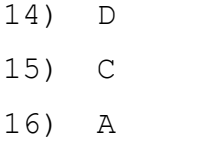
1) C
2) B
3) C
4) C
5) D
6) C
7) D
8) D
9) B
10) B

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

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

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

Question 32 BS tree



Postorder traversal
1316985

40) There are 17 2-hop flights, and ADBCDA is the path for the only 5-hop Austin to Austin round trip.

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.
- **Finally, here is an idea for a scoring option for the last question to award partial credit. Award 3 points for each attempted answer, but penalizing one point if it's wrong. For example, just trying one of the answers and getting it right would add 3 points. Trying two answers and getting one right and the other wrong would result in points added and 1 deducted, a gain of 2. Just a suggestion…** ☺

Brief Explanations:

1. $10100_2 + 10110_2 = 20_{10} + 22_{10} = 42_{10} = 52_8 = 2A_{16} = 101010_2$
2. The == compares object references, false in this case since x and y are pointing to different objects, **compareTo** returns a 0 since the object values are equal, and **equals** returns true, again since the values are equal.
3. There are 7 characters in "riptide".
4. P and P is simply P, whatever it was, true in this case. Q gets the result of true and false, which is false.
5. The switch statement only works with ints, chars, and now Strings. Floating point values cannot be used as cases.
6. To instantiate an int array as designated, you must have the [] and either a static list {0,0,0,0,0} or the new int [5].
7. The first and last sections of a for loop do NOT have to included inside the structure..only the middle Boolean section does. When you do this, it's like doing a while loop with the start happening before the loop, and the step happening inside the body.
8. This Guitar class has only three instance variables. The static variable belongs to the class, and can be accessed by the object, but it is not exclusive to any object instance.
9. The two parameter constructor must be used, since the default values are different. The order is important…int first, then string since that is how the constructor is defined.
10. Each time an object is instantiated, the serial number increments first, so the ukulele gets 101 and the mandolin gets 102.
11. Even though a double is used in the increment, since the -= operator is used, the result is automatically cast back to a float, so there is no data type mismatch here.
12. The absolute value of -56 is 56.
13. The char loop is assigned using ints, which represent the char values of '0' through '9', so the String is built starting on '0' up to but not including '9', resulting in "012345678".
14. This single integer substring call takes the string starting at position 4 all the way to the end of the string, resulting in "nyCricket".
15. The int variable a starts at 10, becomes 11, and then is subtracted by 12, becoming -1.
16. The two's complement binary representation of -1 is simply all 1s across the board, the complement of zero.
17. 927354 % 1000 yields 354, and /10 results in 35. 927354 / 1000 yields 927, and % 10 results in 7.
18. The Boolean result of OR is true when either or both values are true, with **false or false** the only false result.
19. The complement of -100 (opposite, minus 1), is 99, which when right shifted 1 (divided by 2) results in 49.
20. The method **stuff** returns the reverse substring between the given locations, as long as the parameters are in ascending order. The result of ("Simple method", 3, 10) starts with the character in position 9, the 't', and goes backwards to the 'p' in position 3.
21. The ("Rumplestiltskin", 15, 0) call results in an empty string since the parameters are not in ascending order.
22. Despite filling each row of this 3X6 char 2D array with stars, the reassignment using the toCharArray negates that, resulting in no extra elements other than the ones in the strings.
23. On the right see the trace for this recursive call.
24. The string "45" is parsed into base 10, then converted to a base 8 string, becoming "55".
25. The Arrays.binarySearch method returns the index of the found element, or the index of where it should have been, plus one, then times -1. The 3 was first found in position 2, but the 7 is not in the list and should have been in position 7, thus returning (7+1)*-1, or -8.
26. The **"\D"** wild card character indicates non-digit characters, which means splits happen on any non-digit, namely all the letters in this string. Since the "a" is at the front of the list, an empty string is created at the front. However, trailing empties are NOT returned by the split method, as you might think with the "de" at the end of the string. The result is an array with one empty string at the front, followed by "123", "45", and "6789".
27. This is a simple replace method, with 'a's replaced by 'u's and "an"s replaced by "oo"s.
28. Right shifting by 0 is dividing by 2 to the zero power, or 1, resulting in 100. The two's complement binary representation of 100 is 1100100.
29. This simple for each loop outputs the double of each element in the list.
30. The long data type is stored in 64 bits.
31. The digital electronics arrow symbol represents the Boolean OR gate. Just remember that the word "ARROW" has OR in it, even if it is reversed…a good way to remember anyway.
32. In this binary search tree implementation, nodes are inserted either left or right of the root node and any subsequent child, with ties allowed, going to the left. The print is in postorder since the output statement is after the left child and right child recursive calls. See the key for the resulting tree.
33. The average case for inserting a node into a BST is O(log N).
34. This is one is easy…just the data in the first node.
35. The "a*.*" pattern matches zero or more 'a's followed by zero or more of any character. The ".+b+." string matches one or more of any character, followed by one or more 'b's, followed by a single character. The ".*c" matches zero or more of any character, followed by the letter 'c'. All patterns match in these examples, resulting in all trues.
36. The Comparable interface must be "implemented" with the class indicated in <>… **`class Boo implements Comparable<Boo>`**.
37. Since the instance variables are x, y, and z, stating just x in the method implies this.x, which is compared to b.x, the x belonging to the parameter b. All three current object fields must be greater than the corresponding parameter object fields in order for **return 1** to occur.
38. Object a is indeed greater than object b since 2 > 1, 'd' > 'a', and 3.7 > 3.4, returning a 1. Objects a and c are equal since all fields match, returning a 0. Object b is less than c since its fields are less, returning a -1.
39. After the root value 15 is popped, and replaced by the last element in the heap, the 2 is "sifted down" into place, first swapped with the greater of its two children, the 10, then swapped with the 9, finally settling in position 4 of the tree.
40. There are 17 two-hop flights in this system. Here they are: ADA, ADB, BCB, BDB, BDA, BCD, BCC, CCC, CBC, CDB, CDA, CCB, CCD, CBD, DBD, DAD, DBC. There are two ways to find this…brute force as shown above, or matrix multiplication. Brute force works on small graphs like this, but I suggest you learn matrix multiplication for anything more complex. The only 5-hop round trip flight from Austin is ADBCDA.