Computer Science Answer Key
UIL Invitational B 2015

1)	D	11)	E	21)	C	31)	D
2)	E	12)	C	22)	B	32)	B
3)	B	13)	C	23)	C	33)	C
4)	E	14)	B	24)	D	34)	B
5)	B	15)	B	25)	C	35)	D
6)	A	16)	D	26)	A	36)	C
7)	A	17)	A	27)	B	37)	C
8)	A	18)	C	28)	D	38)	B
9)	D	19)	C	29)	C	39)	NOT POSSIBLE
10)	A	20)	A	30)	C	40)	$A * (\overline{B} + \overline{C})$

Note: Since AND and OR have the commutative property, any answer that is a correctly commuted version of this answer is correct.

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.

Explanations:

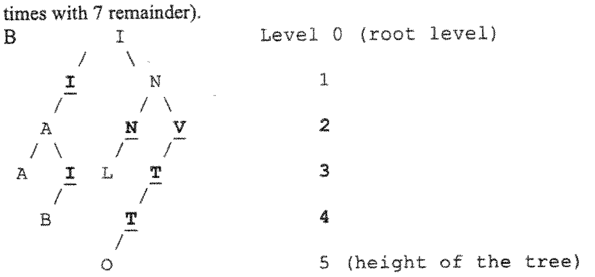
1. D $10111000_2 + AB_{16} = 184_{10} + 171_{10} = 355_{10} = 543_8 = 163_{16} = 101100011_2$
2. E 23 / 9 * 1 % 2 = 2 * 1 % 2 = 2 % 2 = 0
3. B Since a field width of 4 is not enough to contain 49.20, the compiler just takes what it needs and left justifies the output.
4. E The "IE" pattern finds the only "IE" in the word and replaces it with a single "O".
5. B Since p is true and q is false, p and q is false, which makes the NOT of P and Q true.
6. A This a lesser known Pythagorean triple...8, 15, 17. The Math.hypot method returns the hypotenuse for sides of 8 and 15, which when applied to the Pythagorean theorem becomes 64 + 225, or 289, whose square root is 17.
7. A 8 * 1.4 = 11.2, which truncates to 11 in the autocast provided by *=
8. A 14%3 equals 2, which results in a y value of 2 in the switch statement, and an output value of 14+2, or 16.
9. D The trace values of x are 4, 8, 9, 18, 19, 38, 39, 78, and 79, with the output value being 79.
10. A The sequence of the first loop is as follows: position 6 (5+1) of list2 gets the value ++a (2), 4 gets 3, 2 gets 4, 3 gets 5, 5 gets 6, and 1 gets 7. Positions 0 and 7 of list2 remain the default values 0. The second loop simply outputs the values of list2.
11. E For output purposes only, any of these statements will work with this data.
12. C The value sequence for d and x is: 100.0 0, 150.0 1, 225.0 2, 337.5 3, 506.25 4, 759.375 5, 1139.0625 6.
13. C The `expr++ expr--` (postfix) operators are on line 1 of the chart, followed by | on line 10, and || on line 12.
14. B Half of the values are negative (-128 to -1), and the other half non-negative (0-127).
15. B This toArray method of the ArrayList class returns an Object array, therefore the word Object must fill the <statement> in the code.
16. D

A	B	C	$\overline{A}$	$B*C$	$\overline{A}+B*C$	$\overline{\overline{A}+B*C}$
0	0	0	1	0	1	0
0	0	1	1	0	1	0
0	1	0	1	0	1	0
0	1	1	1	1	1	0
1	0	0	0	0	0	1
1	0	1	0	0	0	1
1	1	0	0	0	0	1
1	1	1	0	1	1	0

17. A 65 - 97 is equal to -32, which when tripled equals -96.
18. C The contents of the matrix after the assignment loop are:

```
0 0 45 30 15
0 36 21 6 0
0 12 0 42 27
0 48 33 18 3
0 24 9 0 39
```

19. C 118 in 8-bit binary is 01110110. Applying the rule mentioned in the problem gives you 10001010, which is the binary equivalent of -112.
20. A The worst case running time for a quick sort is O(N^2), which occurs when the list is already sorted, or when only a few elements are not in sorted order.
21. C This quick sort arranges a list of numbers in **descending** order due to the given comparison operators in statements 3 and 4. The output is the choice that shows all of the numbers in order from greatest to least.
22. B The comparison operators in both statements 3 and 4 need to be reversed for the quick sort to work in reverse order than is indicated here.
23. C The output string is `"aeioueiouiouou"`, which contains four letter 'o's.
24. D The tangent of a 45 degree angle is 1.0 (opposite over adjacent)
25. C The value 10 in 8-bit binary is 00001010, which when made negative becomes 11110110 in twos complement form. When right shifted twice, it becomes 11111101, which is the decimal value -3.
26. A A precondition of this binary search algorithm is that the list is in ascending sorted order.
27. B The sorted list is -7, -5, -3, 0, 1, 3, 4, 5, 8, 9. Position 7 contains the 5, and -6 is not in the list, which is indicated by -1.
28. D The pattern ".*" matches zero or more of any character, ".+" matches one or more of any character, and the "?." matches one character, once or not at all.
29. C The recursive call value sequence for each initial call is: 6, 3, 0, result 2+2+1 = 5, 10, 7, 4, 1, -2, result 2+2+2+2+3 = 11, -5, result 3
30. C Since all alpha characters are above 60 in value, and all numeric and space characters are below, "A" is printed for the letters, and "B" is printed for the digits and the space character with this ternary operator statement.
31. D The base ten values of 7, 34, and 63 convert to base 8 values of 7, 42 (8 goes into 34 4 times with 2 remainder), and 77 (8 goes into 63 7 times with 7 remainder).
32. B



There are 6 parent nodes with only one child, shown in bold and underlined, and tree height is 4 (root node is at level zero, and furthest leaf is at level 4). In this binary search tree exercise, duplicates are allowed and are inserted to the left of matching elements. The first node is at level zero, therefore the resulting tree, as you can see in the key above, has a height of 4, with 5 leaves.

33. C The sequence of the first four statments is: call to default constructor(A), call to toString method(F) which returns and outputs the value 5, call to setThing(D) which changes the instance field value to 6, and a final call to the toString method(F), which returns and outputs the value 6.

34. B The sequence of the next two statments is: call to one parameter constructor(H) which assigns 3 to the ThingTwo instance field, but first automatically calls the ThingOne default constructor(A) which assigns 5 to the ThingOne instance field, followed by the call to the ThingTwo getThing method(I), which returns and outputs the instance field value of the ThingTwo object(3).

35. D The sequence of the last two statements is: call to reduceThing(K) which reduces the ThingTwo instance field value to 2, then call to the ThingTwo toString method(L), which in turn calls the ThingOne toString method(F), and returns concatenated values of both instance fields, 5 and 2.

36. C The first rule is that the operands stay in the same order, A B C D E F. Then follow order of operations. The exponent goes in front of the E and F, and the * before CD. The first + goes in front of the AB, then the minus before that, then finally the last + in front of everything.

37. C The signals NOT A and B go into an AND gate, which feeds into a NOT OR gate, which receives an AND signal from C and NOT D. The NOT OR result goes into an AND gate which also receives the A signal.

38. B See a complete tracing of this recursive function call shown above.

$g(-10) = g(-5)+g(-2) = 0+1 = \boxed{1}$

$g(-5) = g(0)+g(3) = 1+-1 = 0$

$g(0) = g(5)+g(8) = -1+2 = 1$

$g(3) = -1$

$g(-2) = g(3)+g(6) = -1+2 = 1$

$g(5) = -1$

$g(8) = 2$

$g(6) = 2$

39. No matter where you start in this graph, it is **not possible** to cross all seven bridges without recrossing any of them. However, adding an 8th bridge, perhaps another between C and D, would enable an Euler path such as this: BADABCDCA. This problem helped give birth to the field of graph theory the development of which Euler is given much credit.

40.

$A * (\overline{B} + \overline{C})$

Explanation: In the diagram below, the distributive property is used to simplify the original expression. Absorption is used by the second term to eliminate the third term, and then the A is factored out, resulting in the final expression, which contains two operators, AND and OR, with NOTs over both the B and C terms.

$$A(\overline{B}+\overline{C}) + A(\overline{B+C})$$

$$= A\overline{B} + A\overline{C} + A\overline{B}\,\overline{C}$$

$$= A\overline{B} + A\overline{C}$$

$$= A(\overline{B}+\overline{C})$$