

# ★ANSWER KEY – CONFIDENTIAL★

## UIL COMPUTER SCIENCE WRITTEN TEST – 2016 INVITATIONAL A

Questions (+6 points for each correct answer, -2 points for each incorrect answer)

- |                  |                  |                  |                     |
|------------------|------------------|------------------|---------------------|
| 1) <u>  C  </u>  | 11) <u>  C  </u> | 21) <u>  A  </u> | 31) <u>  B  </u>    |
| 2) <u>  A  </u>  | 12) <u>  D  </u> | 22) <u>  E  </u> | 32) <u>  D  </u>    |
| 3) <u>  D  </u>  | 13) <u>  B  </u> | 23) <u>  D  </u> | 33) <u>  C  </u>    |
| 4) <u>  B  </u>  | 14) <u>  C  </u> | 24) <u>  C  </u> | 34) <u>  E  </u>    |
| 5) <u>  A  </u>  | 15) <u>  E  </u> | 25) <u>  A  </u> | 35) <u>  A  </u>    |
| 6) <u>  A  </u>  | 16) <u>  C  </u> | 26) <u>  C  </u> | 36) <u>  D  </u>    |
| 7) <u>  E  </u>  | 17) <u>  C  </u> | 27) <u>  D  </u> | 37) <u>  B  </u>    |
| 8) <u>  D  </u>  | 18) <u>  D  </u> | 28) <u>  C  </u> | 38) <u>  D  </u>    |
| 9) <u>  C  </u>  | 19) <u>  C  </u> | 29) <u>  C  </u> | 39) <u>HDBCFL</u>   |
| 10) <u>  A  </u> | 20) <u>  B  </u> | 30) <u>  A  </u> | 40) <u>  A+C  *</u> |

\* See "Explanation" section below for alternate, acceptable answers.

**Note:** Correct responses are based on **Java SE Development Kit 8 (JDK 8)** from Sun Microsystems, Inc. All provided code segments are intended to be syntactically correct, unless otherwise stated (e.g., "error" is an answer choice) and any necessary Java SE 8 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used.

### Explanation

- 1) C  $123_8 + 45_8 = 78_{16} = 170_8 = 1320_4 = 120_{10}$ . Note that answer B ( $168_8$ ) is not a valid octal representation.
- 2) A  $x = 6.0, y = 0.0$  (integer division),  $z = 6 * 0 = 0.0$
- 3) D "%d pts" formats a string consisting of a decimal representation of right (6) followed by the string literal " pts". Variables wrong, skip, and pts are not used in the output.
- 4) B "[cabbage]": Regex matches on any 'a', 'b', 'c', 'e', or 'g' (redundant 'a' and 'b' in regex are ignored).
- 5) A  $a < b \rightarrow r = \text{false}, a == b \rightarrow r = \text{true}; a > b \rightarrow r = \text{true}$
- 6) A `bang = 4.0` (`ceil()` rounds up), `pow = 3.0` (`floor()` rounds down), `oof = 3.0` (`min()` returns the minimum parameter)
- 7) E `nibble -= nibble` results in `nibble` being assigned a value of 0. This later leads to `nibble /= nibble`, which results in a "Divide by Zero" exception.
- 8) D Both `if()` conditions are independent of one another, so the 2<sup>nd</sup> `if()` will always be evaluated, regardless of the outcome of the 1<sup>st</sup> `if()`. The else clause is not evaluated here since it is contingent upon the 2<sup>nd</sup> `if()` evaluating to false, which it doesn't.
- 9) C `ch` iterates through every other uppercase letter from 'F' through 'L' (exclusive). Each character literal is printed in succession, all on the same line of output.

# ★ANSWER KEY – CONFIDENTIAL★

- 10) A `alpha[5] = alpha[3]--;`: The post-decrementation operator (`--`) assigns the original value of `alpha[3]` (e.g., 12) to `alpha[5]`. Then `alpha[3]` is decremented from 12 to 11.
- 11) C `total = -10 + 8 + 7 = 5`: Each iteration of the loop reads of 2 integers from the input string. If the first integer is even, the second integer is added to `total`. The loop exits when the Scanner reads off the 7.
- 12) D This code accumulates powers of 2. At the end of each iteration, `seqA` and `seqB` are always powers of 2. The loop exits when the accumulated sum reaches 63.
- 13) B `r = 0 + 2 = 2` (side effect: `q = 2`). `s = 2 + 4 = 6` (side effect: `q = 4`).
- 14) C `short` and `char` = 16 bits of memory
- 15) E "Dewey" inserts before "Donald", "Louis" inserts before "Huey", "Daffy" replaces "Donald"
- 16) C  $511_{10} = 011111111_2 \gg 3 = 000111111_2 = 63_{10}$ . Also  $511 / 2^3 = 511 / 8 = 63.875 \rightarrow$  truncates to 63.
- 17) C  $999999_{10} = 1212210202000_3$
- 18) D table: 

0	1	2	3	4
1	2	3	4	5
2	3	4	5	6
3	4	5	6	7

 $0 + 1 + 2 + 3 + 4 = 10$  rows: 

10
14
15
13

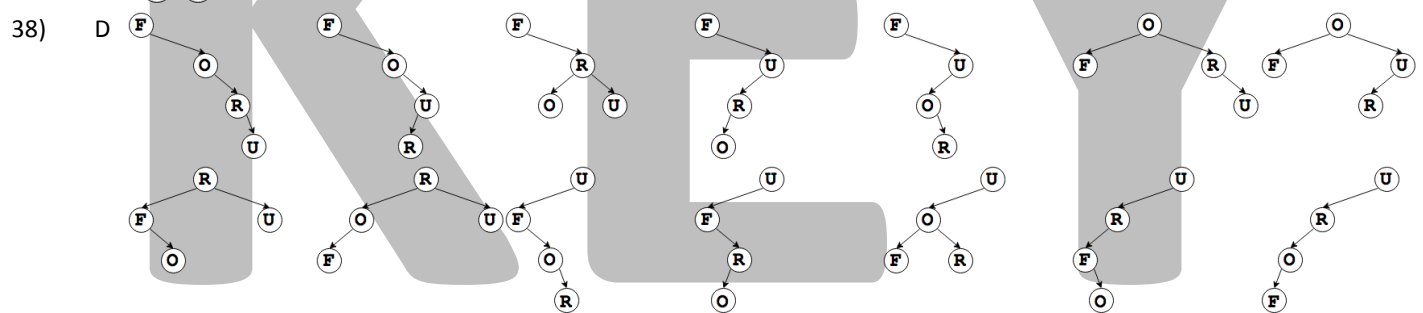
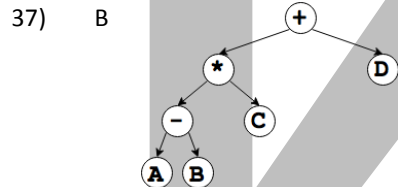
 $2 + 3 + 4 + 5 = 14$   
 $4 + 5 + 6 = 15$   
 $6 + 7 = 13$
- 19) C Strings are sorted lexicographically by Unicode character values. "null" is a String literal, not the null reference.
- 20) B Alpha is abstract and cannot be instantiated as a new `Alpha()`. Constructor `Beta(String id)` in class Beta requires a String parameter and cannot be instantiated as a new `Beta()`.
- 21) A Since `agent` is instantiated as a new `Beta("007")`, the overridden method `id()` in the Beta class is used, which accesses the instance variable `id` privately declared within the Beta class (i.e., initialized w/ "007").
- 22) E The `toString()` method is inherited from the Alpha class, which accesses the instance variable `id` privately declared within the Alpha class (i.e., initialized w/ "Echo").
- 23) D  $O(1) < O(\log_2 N) < O(N) < O(N * \log_2 N) < O(N^2)$
- 24) C Regular expression `"\w+\d+"` (encoded as a String w/ escape characters: `"\\w+\\d|"`) means "one or more word characters (i.e., `\w = [a-zA-Z_0-9]`) followed by one or more digit characters (i.e., `\d = [0-9]`)".
- 25) A Each true value in the matrix correspond to a directed link from the node for the column to the node for the row.
- 26) C Links are represented by non-weighted Boolean values. The matrix is not symmetric (i.e., a directed graph). Nodes D and E are disconnected from nodes A, B, and C.
- 27) D
- | P | Q | !(P && Q) | Q | P | Q | !P && !Q | !P | Q |
|---|---|-----------|---|---|---|----------|----|---|
| 0 | 0 | 1         |   | 0 |   | 1        | 1  |   |
| 0 | 1 | 1         |   | 1 |   | 0        | 1  |   |
| 1 | 0 | 1         |   | 1 |   | 0        | 0  |   |
| 1 | 1 | 1         |   | 1 |   | 0        | 1  |   |
- 28) C String concatenation results in the String "510" to be parsed into an integer.
- 29) C *Quicksort*:  $O(N^2)$  in worst case. *Sequential Search*:  $O(N)$  in worst and average cases,  $O(1)$  in best case. *Binary Search*:  $O(\log_2 N)$  in worst and average cases,  $O(1)$  in best case. *Selection Sort*:  $O(N^2)$  in all cases.
- 30) A String splits on all individual lowercase vowels. Empty string at the end of the array after the final "e" is truncated.
- 31) B Each recursive pass appends a "splat" (#) and `n` trailing dots to the end of the resulting string. Base case: `n <= 0`.
- 32) D `stack = [10, 24, 36, 11, 9, 7] = [10, 11]`
- 33) C `queue = [(36 + 24), (7 - 9)] = [60, -2]`
- 34) E Java equivalent: `(!P || Q) && !R`

# ★ANSWER KEY – CONFIDENTIAL★

35) A

P	Q	R	$(\overline{P} + Q) \cdot \overline{R}$
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

36) D  $55_{10} = 00110111_2$



39) Pre-order: HDBCFL. Post-order: CBFDLH. In-order: BCDFHL. Level-by-level: HDLBFC.

40) Any answer that equivalently expresses "A Logical-OR C" is acceptable (e.g., " $A + C$ ", " $C + A$ ", " $A || C$ ", " $C || A$ ", " $A$  or  $C$ ", " $C$  or  $A$ ")