★ANSWER KEY – CONFIDENTIAL★

UIL COMPUTER SCIENCE WRITTEN TEST – 2016 STATE

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

1) D	11) C	21) C	31) D
2) A	12) E	22) B	32) A
3) C	13) A	23) A	33) E
4) E	14) C	24) C	34) C
5) B	15) B	25) D	35) B
6) B	16) A	26) C	36) D
7) A	17) A	27) A	37) D
8) E	18) D	28) B	38) B
9) B	19) E	29) C	39) –256
10) B	20) A	30) C	40) 60

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) D 23_5 * 13_7 = 11211_3 = 10002_4 = 1010_5 = 334_6 = 244_7

2) A `y = 7.5, x = 7 * 2 = 14` (type-casting to `int` applies only to `y`, not `y * 2`)

3) C `"[%-7.3f]\n"`: Round the parameter to 3 digits of precision and left-justify within a 7-character wide field bounded by square brackets and terminated with a newline character. `"[%7.3f]\n"`: Round the parameter to 3 digits of precision and right-justify within a 7-character wide field bounded by square brackets and terminated with a newline character.

4) E The regular expression `"."` matches on *any* character, not just the literal period (`"."`).

5) B

P	Q	R	**X**	A)	B)	C)	D)	E)
0	0	0	**1**	1	1	1	1	1
0	0	1	**0**	0	0	0	0	0
0	1	0	**1**	1	0	1	1	1
0	1	1	**1**	1	0	1	1	1
1	0	0	**0**	0	1	0	0	0
1	0	1	**0**	0	1	0	0	0
1	1	0	**0**	0	0	0	0	0
1	1	1	**1**	1	1	1	1	1

6) B After first loop, `data = {12, 9, 11, 11, 12, 12}`

★ANSWER KEY – CONFIDENTIAL★

7) A Maximum value for a byte is `127`. Therefore, `60 * 3 = 180` results in overflow. The 8-bit, unsigned integer representation of `180` is equivalent to the 8-bit, 2's complement representation of `–76`.

8) E An `ArithmeticException` (divide by zero) is thrown when `j / k` is evaluated. The use of logical OR (`||`) does not lead to short-circuit evaluation in this instance.

9) B Iterates backward through the `String`, comparing `'z'` with `'Z'`, `'Y'` with `'Y'`, `'x'` with `'x'`, and `'W'` with `'w'`.

10) B Contents of `animal` after each iteration of the `for()` loop:

```
[bird, cat, cow, dog, fish, horse]
[bird, cow, dog, fish, horse, pear]
[bird, cow, fish, horse, pear, plum]
[bird, cow, fish, horse, pear, plum]
[bird, cow, fish, horse, pear, plum]
[bird, cow, fish, horse, pear, plum]
```

11) C The `if()` condition reads and tests a token from the input (e.g., `"This"`). If the condition evaluates to true (i.e., the token's length is 4), the *next* token (e.g., `"is"`) is read and converted to uppercase (e.g., `"IS"`) to be printed.

12) E `total = 22 + 19 + 16 + 13 – 10 + 7 + 4 + 1 = 72`

13) A
```
= ((3.0 != 3) ? (2 ^ 3) : (3 / 4))
= (false ? (2 ^ 3) : (3 / 4))
= (3 / 4)
= 0
```

14) C 52 characters @ 2 bytes (16 bits) per `char` = 104 bytes

15) B
```
one = [3, 3, 5, 4, 4, 3];
two = [one, two, three, four one two six, five four five, six three]
```

16) A From Section 15.17.3 of the Java Language Specification, "the result of the remainder operation can be negative only if the dividend is negative, and can be positive only if the dividend is positive" and that `a % b = a – b * (a / b)`

17) A The `check()` method calculates the parity bit for even parity in `input`.

18) D `get()` performs binary search on `String z`, returning the index position where the search term, `y`, is located or where it would be located if it is not in the string. `"Aegilops"` does not contain `'m'`, but alphabetically, `'m'` should be located at index 5 between the `'l'` and the `'o'`.

19) E Addition has higher precedence than bitwise left shift (i.e., `i << (i + 1)`).

20) A Appending a list to itself is the same as appending any other singular object to the list (i.e., `size()` increases by 1).

21) C `[0–9]` = digits 0 through 9. `[^0–9]` = non-digits (e.g., upper- and lowercase letters, space, comma, slash, etc.)

22) B `sort()` returns the maximum depth of recursion (i.e., $\log_2 14 = 3.807$, rounds up to 4).

23) A Sorts the elements of the list into descending order.

24) C Recursively sorts the two halves and merges them together into a sorted list.

25) D Merge Sort yields a O(N * $\log_2$ N) performance in the best, average, and worst cases.

26) C `'e'+1+"e+1" = (101 + 1) + "e+1" = "102e1"`, parsed as scientific notation (i.e., 102×10^1).

27) A `matrix` is filled in column-major order. `matrix[3][2] = 'I' = 73`. `matrix[2][1] = 'E' = 69`.

28) B Bitwise XOR (exclusive OR).

29) C The class variable `x` is initialized to 2 before first constructor increments it to 3.

30) C The `x` parameter in the 2nd constructor shadows the class variable `x`. As a result, the statement `x += x` does not affect the value of the class variable.

31) D Elements of a stack are removed in the reverse order from which they are added to the stack.

32) A Elements of a queue are removed in the same order in which they are added to the queue.

33) E Elements of a priority are removed in ascending order regardless of the order in which they are added to the queue.

34) C Variable `y` references the instance variable of the subclass (`"Why"`). The `why()` method returns the instance variable of the superclass (`"Y"`).

35) B `super.y()` references the overridden `y()` method of the superclass, which returns the instance variable of the superclass (`"Y"`).

★ANSWER KEY – CONFIDENTIAL★

36) D $(X * Y * Z) + (\overline{W} * X * Z) + (W * X * \overline{Y} * Z)$

$(X * Z * Y) + (X * Z * \overline{W}) + (X * Z * W * \overline{Y})$ Commutative Law: A * B = B * A

$((X * Z) * Y) + ((X * Z) * \overline{W}) + ((X * Z) * (W * \overline{Y}))$ Associative Law: A * B * C = A * (B * C)

$(X * Z) * (Y + \overline{W} + (W * \overline{Y}))$ Distributive Law: (A * B) + (A * C) = A (B + C)

$(X * Z) * (Y + \overline{W} + \overline{Y})$ Law of the "Disappearing Opposite": A + ($\overline{A}$ * B) = A + B

$(X * Z) * (\overline{W} + Y + \overline{Y})$ Commutative Law: A + B = B + A

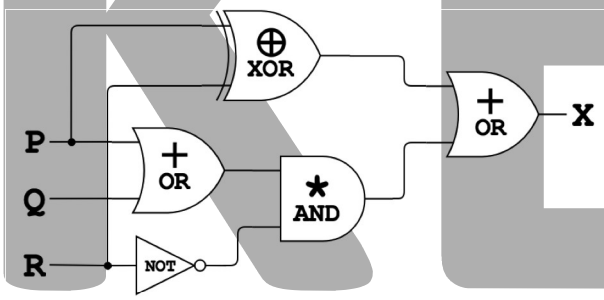
$(X * Z) * (\overline{W} + 1)$ Law of Complement: A + $\overline{A}$ = 1

$(X * Z) * (1)$ Law of Union: A + 1 = 1

$(X * Z)$ Identity Law: A * 1 = A

37) D Longest path: F → C → A → D → B

38) B



39) 2's complement equivalent of 1111 1111 1111 1111 1111 1111 0000 0000 = –256.

40) (9 – 7) * ((8 – 3) + (((6 + 4) / 2) * 5)) = 60