

Computer Science Contest #1415-01 Key

October 11, 2014

- |       |               |
|-------|---------------|
| 1) B  | 21) A         |
| 2) D  | 22) C         |
| 3) A  | 23) B         |
| 4) B  | 24) C         |
| 5) D  | 25) C         |
| 6) D  | 26) E         |
| 7) C  | 27) D         |
| 8) D  | 28) A         |
| 9) C  | 29) A         |
| 10) A | 30) E         |
| ■     | ■             |
| 11) A | 31) D         |
| 12) A | 32) C         |
| 13) C | 33) E         |
| 14) B | 34) C         |
| 15) B | 35) B         |
| 16) A | 36) D         |
| 17) D | 37) C         |
| 18) B | 38) D         |
| 19) A | 39) -82       |
| 20) C | 40) 1010 0001 |
| ■     | ■             |

**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.  $239_{10} + A_{16} = 11101111_2 + 1010_2 = 11111001_2$
2.  $95/13 = 7$  (DIV),  $95\%13 = 4$  (MOD),  $7-4 = 3$
3. `%3f` means to go .XXX of the double provided, `%s` means to print the String
4. `substring(x,y)` will get the string at index x all the way up to, but not including, index y. So in this case QUES is grabbed, but not the T
5. `&&` is always evaluated before `||`, this question assumes you got question 4 correct.
6. `Math.random()` ->  $[0,1)*17$  ->  $[0,17)$  -> `(int)[0,17)` ->  $[0,16]-6$  ->  $[-6,10]$
7.  $\text{sum} = 6 + 12 + 18 + 24 + 30 + 36 + 42 + 48 + 54 = 270$
8. "the" is under the case of 'O', which is at `charAt(3)` in "AEIOU"
9. 

i	->	X	1	2	3	4	5	6	7	8	9		
		X	->	5	0	1	1	0	3	0	3	6	0
10. `list[1]` -> 4, `list[4]` -> 0. You should always work from inside out.
11. the integer 4 and the double 4.6 will both be accepted by `nextDouble`, but "4.6" will be an item mismatch (because of the "")
12. i will be 2, 6, 10, 14, and 18. Add of that to 17 and you get 67
13. multiplicative comes before shift, assignment always comes last, relational comes before logical
14. A is a short, C is a byte, D is a double, E is a float.
15. 

loop 1:	y=5,	x=0,	swap x->y-1	list=5,2,3,4,1	x = 2,	y = 4
loop 2:	y=4,	x=2,	swap x->y-1	list=5,2,4,3,1	x = 1,	y = 3
loop 3:	y=3,	x=1,	swap x->y-1	list=5,4,2,3,1	x = 2,	y = 2
loop 4:	y=2,	x=2,	swap x->y-1	list=5,2,4,3,1	x = 1,	y = 1
loop 5:	y=1,	x=1,	swap x->y-1	list=2,5,4,3,1	x = 0,	y = 0

 end loop
16. the default constructor does not exist because a user defined constructor was written. In order to have a default constructor, you must write one.
17. words cannot be used outside of the class because it is a private member. `toString` does not accept a parameter.
- 18&19. the matrix is filled up in row major order (fill the row first) and the String is built by column major order(fill the column first).
20. `%x` means to print in hexadecimal. You must perform all `<<` before you do the `|` which will yield 101011111100 rather than 101010100011111100.
21. This is not a true sort, be wary that j is being compared to i and that j only increases when a swap occurs.
22. If the alphabetically last word is at the beginning of the list then every word will be compared to that word and no change will occur
23. P is a W because P is a U and U is a P. Therefore P has all the characteristics of W.
24. Since the parent class does not have a default constructor, the child class must call `super` first.
25. you cannot call `super()` alone, you must call `super.toString()`;
26. Even though you checked to see if the object is a Bonus, you did not cast that Candy object into a Bonus.
27. Queue is an interface and must be constructed by a List class. `element` and `peek` do the same thing as `remove` and `poll`.
- 28&29. The way the recursion works, it is the difference of x and y as long as both values are above 0, if either value is below zero then you treat the value as if it was zero.
30. `^` means bitwise xor and it takes the binary value of two numbers and returns a 1 if each corresponding bit is opposite. Mathematically, if the number is bigger than 32, then the result will be the difference while if the number is less than 32, then the result will be the sum.
31. In a pre-order traversal, the root is printed, then you go left, then you go right.
32. The stack follows the first in/last out rule, which means you will not go on to the next String until the String you are currently working on is empty.

33. Set contains no duplicates and TreeSet is in alphabetical order. So it will be the first five unique alphabetical combinations.

34. The best way to find this is to take out the duplicate letters.

35. because the String starts with "a", there is an empty string created for the first element of the array.

36. The UIL will now concentrate on Boolean Algebra. In this question, we use DeMorgan's Law to solve  $!(A+B) \rightarrow !A!B \rightarrow A!B$

37. The UIL will now concentrate on Digital Electronics. The D symbols represent AND the Curved D with the point is an OR. The 3 quadruples are

A	B	C	D
1	1	0	1
1	1	1	0
1	1	1	1

38. The UIL will now concentrate on prefix, postfix, and infix expressions. A postfix expressions has the operation after the operands. So you do this:

$$D+B*(F-E/C)*(D-A*B)$$

$$\begin{array}{l} EC/ \quad AB* \\ FEC/- \quad DAB*- \\ BFEC/-* \quad DAB*- \\ BFEC/-* \quad DAB*- //These two operands are ready to be multiplied \\ BFEC/-*DAB*-* \\ DBFEC/-*DAB*-+ \end{array}$$

39. The UIL will now have its last two questions as free response questions. The UIL will now be concentrating on 2s compliment which is how you determine this value: Since the first digit of the byte is 1, and it is a signed data type, this indicates that it is a negative number. To determine which negative number you perform a bitwise NOT on it and adding 1.

$$10101110 \rightarrow 01010001+1 \rightarrow 01010010 \rightarrow 81 \text{ which means it is } -81$$

40. The UIL will now be concentrating on bitwise theory, operations, shifts, and circles. A shift will shift bits in the desired direction and replace the open area with 0s, a circle will rotate the bits.

LCIRC-5(RSHIFT-4 11011010) will shift over by 4 to the right

LCIRC-5(00001101) will rotate left 5 bits, and not lose the bits but place Them back on the right

10100001