A+ Computer Science Contest #2526-08 KEY

January 17, 2026

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

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

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

31) A
32) C
33) D
34) B
35) B
36) E
37) B
38) C
39) 01111001
40) 6

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 20 Packages are assumed to have been imported as needed.
- Assume any undefined (undeclared) variables have been defined as used.

© A+ Computer Science – www.apluscompsci.com

1.	A	20313_4 is 567_{10}, AB_{12} is 131_{10}, $567_{10}-131_{10} = 436_{10}$. 436_{10} is $1B4_{16}$. Which means $1B0_{16}$ is not equivalent. All the other values are equivalent to 436_{10}.
2.	E	The && is logical AND. It cannot be used in this equation. If it was the bitwise &, then the answer would have been 72.
3.	B	%% will print a %. %6.5f will set aside 6 spaces to print the float and that float will go to the 5th decimal place, even if it goes beyond the 6 spaces. Java places a 0 in front of a decimal place when printing a decimal number. \\b prints \b which avoids using the escape sequence \b. %d prints out the integer.
4.	B	[Aa] means the delimiter can be either A or a. When splitting a String, if the first character is a delimiter, then the 0th index of the list will be an empty String. If 2 delimiters are side by side, then an empty String will be placed in the list. Hence, the first 8 elements of the resulting 9 element array are empty Strings.
5.	B	There is only one combination that results in false: a is true, b is false, and c is false.
6.	D	Math.ceilDiv will return the rounded up value of 31/7 = 5. Math.pow will take the x value and cube it and return it as a double. $5^3 = 125.0$.
7.	E	The unary post-increment has higher precedence than the unary pre-increment and unary plus. x+++x+++++x+x++ would handle the unary post-increments first 5 +6 +++x+x++ Now the expression has a 6++ occur which is not a valid operation. At compile time, java will recognize this error and will not compile because of the syntax. If the code used parenthesis or spaces, the code would have worked x+++x++ + ++x+x++ 5 +6 + 8+8 =27
8.	D	When the switch case is found, all the code to the right of the case is done, until break is encounterd. For case 4: that means STRG, CRIT, and WEAK are appended to the str. A blank line is ignored.
9.	C	97 is the ASCII value of 'a'. Even though you run out of alphabet does not mean it is a runtime error, it will print out the ASCII symbol of that number. In the case of 123, that value is {.
10.	E	The code compiles because every line is syntactically correct. However, when the code is executed, each index is out of bounds for the existing array causing a runtime error.
11.	E	The scanner is reading one value to check the while loop and reading the next value to add to count. This means it will never get to a negative value for the check and runs out of elements to read and gives a no such element exception runtime error.
12.	C	12%2 + 9%4 + 25%3 + 93%8 + 98%2 + 32%4 + 22%3 = 0+1+1+5+0+0+1 = 8
13.	D	The order of operation is --, *, <<, then ^. 41 ^ 2 << 3 * 2 = 41 ^ 2 << 6 = 00101001 ^ 10000000 = 10101001 (or 41+128) = 169

14.	C	Consider Integer.MIN_VALUE to be x. x-=x is zero. Adding Integer.MAX_VALUE will get -1 because there is one more unit of negative numbers than there are positive numbers (0 is not a positive number). Subtracting by MIN_VALUE/2 is the same as adding by MAX_VALUE/2 which results in half of MIN_VALUE.
15.	D	The ? is a wildcard in generic progamming and can be useful when used as actual type paramenter when the programmer doesn't know what generic is being sent up. ArrayList<Object> does not compile because neither List<String> nor List<Object> are subtypes of each other. The programmer could have written ArrayList with no generics and it would have worked.
16.	D	This class sorts on the compareTo where the number of . determines what comes first, from least to greatest, if the String has the same number of . then the value of the number after the last . is used to determine order, from least to greatest.
17.	D	This is using lambda and stream. x.contains(".11") will yield the desired result. Answer B does not because it will also give 42115 since the index of . is -1. x.sum does not exist. C will not include 1.1150.9.2. A will return every line.
18.	C	This is another use of lambda. The block code determines what is filtered in the ArrayList, in this case it keeps all elements that have at least 3 decimals. Then a sort is called, thankfully Dewey was written as a Comparable. Then it printed each line.
19.	C	?: is the ternary operator which is short hand for if else. Every line compiles, but each line goes into none alphanumeric values except C which modulates the x by 36 and treats the very first numbers as the alphabet.
20.	B	The :: is a method referencing operator that calls the method doStuff for each duck. forEach is a lambda call that goes to each duck. List.of is a way of building a List. When each duck is created, the static c increases by 1, so c is at 5 when the first doStuff occurs, so the sixth letter is removed from the first object.
21.	D	The recursion is constantly growing k by one, which means the memory stack will run out of memory before the value k can grow so big that it will become a the smallest integer possible and go towards the value 15.
22.	D	m(20,10) = 10 + m(19,12) = 10 + 12 + m(18,14) = 10 + 12 + 14 + m(17,16) = 10 + 12 + 14 + 16 + m(16,18) = 10 + 12 + 14 + 16 + 2 + m(15, 20) = 10 + 12 + 14 + 16 + 2 + 5 + m(14, 22) = 10 + 12 + 14 + 16 + 2 + 5 + 2 = 61
23.	E	GORDON becomes DON which becomes ON which becomes N. It will never become an empty String. LIGHTFOOT becomes TFOOT which becomes OOT which becomes OT which becomes T. It will never become an empty String. Since neither base cases occur, the recursion continues until a stack overflow.
24.	B	An interface is implemented, a class is extended.
25.	A	The super call must come first in the body of the constructor. There is no default constructor, so D cannot work.

26.	D	super.getRoute() calls the super classes getRoute method. If getRoute() is called, the user will enter into an infinite call of getRoute as it is calling the current method itself. this.getRoute does the same thing. Connection and Route will not work because getRoute is not a static method.
27.	D	Although the list is held as a Route, each Route has been defined and each Route will use their respective Object's getRoute.
28.	A	This is regex, some finer points is ()? means zero or one of the inclosed group may appear. Which means System.System.out.print; results in a false statement but out.println would result in true.
29.	B	The Deque interface is preferred over both the Stack class and Queue interface because it can behave as either a Stack or a Queue. When it acts as a queue, the front of the queue is at index 0 and the back of the queue is at index length-1. All the listed methods are do function in the Deque interface. offer, add, and addLast all put the element at the back of the queue where it belongs.
30.	E	The Deque interface holds stack with the top of the stack at the first index. So push and addFirst will add the element in the first index. add and addLast adds the element to the last index which violates how a Stack works.
31.	A	Once the first four elements of the stack entered, the stack is [PTL, NYC, DEN, LON]. When the two pops occur it becomes [DEN, LON]. The next two elements become [OKC, DAL, DEN, LON] and then the top is removed to become [DAL, DEN, LON]
32.	C	System.arraycopy will copy a portion of the first array to a portion of the second array. In this case, copy nums starting at index 1 and replace the elements in vals starting at index 2 with the next 3 elements in nums (85, 84, 71).
33.	D	The add method does not exist in the Map interface, neither does push (Stack and Deque) nor offer(Queue and Deque)
34.	B	map.keySet() is the method that returns the set of key values that will allow the user to get each pertinent value. map on its own would return a String (the implied toString call) at best. keySet cannot be called without being attached to the object it is part of and map.valueSet() does not exist. map.values() does and would return the values, which could have worked but was not a choice.
35.	B	The keySet is {37,40,66,93} which will print each respective value 72, 45, 19, 63.
36.	E	Once 37 is mapped to 72, the map has increased in size and causes a concurrent modification error and stops the run. The map does not get to print its current look. This would have worked if the programmer would use an iterative loop, but enhanced loops do not work with concurrent modification.
37.	B	A single underscore is considered a keyword and cannot be used as an identifier, however two underscores are considered a legal identifier. A single $ is considered a variable name. Generics are encouraged, but are not necessary, although they will be referenced as Objects and casting will be necessary.
38.	C	k starts at 99 t is 99,33,11,3,1 k/t is 1,3,9,33,99 k becomes 91 t is 91,30,10,3,1 k/t is 1,3,9,30,91 k becomes 83 t is 83,27,9,3,1 k/t is 1,3,9,27,83 k becomes 75 t is 75,25,8,2 k/t is 1,3,9,34

39.	01111001	Two's complement is used to represent a signed integer, where 0 indicates positive and 1 represents negative in the first bit. The two's complement of a positive number is the same as absolute binary representation of the number.
40.	6	The height of a binary search tree is the number of edges connecting the root to the lowest node