

## HOMEWORK 4

Question	Answer	Explanation
1	B	Three dots (...) are using for a method parameter of type varargs.
2	B	Array indexes in Java are starting with zero so to get the first element, f[0] must be used.
3	D	There are two arrays in this question. These are lowercase and uppercase arrays. All arrays are objects not primitives so there are no primitives in here.
4	C	The array braces can be usable before or after the variable name but they are not usable before the data type. double[] tiger and double bear[] are correct usage but []double lion is wrong usage.
5	C	A varargs parameter can be either an array or individual values but an array parameter can only be an array.
6	A	To determine the number of elements in an array, busses.length can be used.
7	C	To create an empty two-dimensional array with dimensions 2x2, int[][] blue = new int [2][2] must be used.
8	B	The code compiles because there is no wrong. Seven element array is created and these elements are showed in order from 0 to 6 with for loop.
9	B	To do searching on arrays, Arrays.binarySearch() method is using. To do sorting on arrays, Arrays.sort() method is using.
10	B	The array's elements' types are String so alphabetical order is using. The output is [1, 10, 9].
11	B	The first index of an array is always zero and the last index of an array is always array's length - 1.
12	C	The size cannot be defined separately when using an array initializer so tiger and ohMy are not correct. If you don't use an array initializer, the size is required. An empty array initializer is allowed so lion and bear are correct.
13	B	A size is required because no elements are being provided when creating arrays. float[] tiger = new float[1] is correct answer because the size and syntax are correctly defined.
14	C	The array must be sorted before calling Arrays.binarySearch() to get an accurate result.
15	A	An array does not expand automatically when it is full.
16	C	Two dimensional array of size 1x2 is created by the code. Lines m1 and m2 assign values to both elements in the outer array. Line m3 try to reference the second element of the outer array but it throws exception because there is no such position.
17	B	Array's elements are sorted alphabetically by array.sort(os) method. "Mac" is the second element of the array so output will be 1.
18	A	The first line must be char[] ticTacToe = new char[3][3] because a multi-dimensional array is created with multiple sets of size parameters.
19	B	Three objects are created when running the code. The first line creates one object; the array itself. The second line and the third line creates two objects totally and points two of the array references to it.
20	B	The braces must be after the type for one-dimensional array so [] String alpha and [] String beta are incorrect. The braces can be used before or after the variable name for multi-dimensional array so String [][] gamma, String[] delta[] and String epsilon[] are correct.
21	B	Option B shows that there are three different arrays of different lengths so it is the correct one.
22	D	The code throws an ArrayIndexOutOfBoundsException because the last element of the array is names.length - 1 not names.length.
23	C	It is not an ArrayList so we should use days.length not days.size so the code does not work.
24	C	Both bools and moreBools can reference a three dimensions array because the braces are used before the variable names.
25	C	Calling toString() on an array does not output the contents of the array so option C is correct answer.
26	B	ticTacToe is 3x3 array so valid indexes are 0,1 and 2.
27	D	The code does not compile because the syntax is invalid. varargs is just used as a method parameter.
28	D	The code compiles but throws a different exception at runtime. Integer value is assigned to a cell in a 2 dimensions array in line 6. It casts to a general Object[] in line 7. The compiler cannot protect us from assigning a String to the int[] because the reference is more generic. Therefore, ArrayStoreException will be thrown because of the incorrect type.
29	C	There is no element "RedHat" in the array. If it is, it would be between the second and third element. If it is not, it takes the negative index. Therefore output is -3.
30	B	The first argument of the array is Wolfie so output will be "Wolfie".

31	C	Output will be 2 because there are two arguments.
32	B	Output is 1 because seed is the second element in the sorted array.
33	D	Option D's array of length is 1 and that element is of length 2.
34	C	When arrays are indexed, numbers are using not strings so dimensions[2][2] is the right answer to access the value of "z".
35	D	The code throws an ArrayIndexOutOfBoundsException because <= operator is used instead of the < operator.
36	C	The code throws an ArrayIndexOutOfBoundsException because there is only one argument is "Wolfie".
37	D	The code compiles without error. 2D array is created in line r1. Values are assigned to an array element in next three lines.
38	D	The code does not compile because array do not have length() method.
39	B	boolean[][] applies to both variables. Then bools gets another dimension from the braces right after the variable name but moreBools stays at 2D.
40	B	No arguments are passed from the command line so Empty array [] is the result when called as java counting.Binary
41	D	Sorting the array is must before binarySearch so the output is not defined.
42	B	The code does not compile because Line 8 attempts to store a String in array meant for an int.
43	A	Output of listing.length is 2 and output of listing[0].length is 1.
44	C	The code throws an ArrayIndexOutOfBoundsException because there is no argument. FirstName is the name of the class.
45	A	Output will be 6 lines because for loop is started from 1. Loop will start to output from "Monday" to "Saturday".
46	B	The output is 1 because there is only one argument. The argument is a String with three characters. These are 1, a space and 2.
47	A	Output is 0 because "Linux" is the first element of the array. Normally, array must be sorted but there is no need in this example because it is already sorted.
48	A	An array parameter and varargs parameter are same in the method so you can always change a method signature from call(String[] arg) to call(String... arg) without causing a compiler error in the calling code.
49	B	All of the variables point to a 4D array but nums2b points 3D array.
50	C	The code does not compile because an int cannot be stored in a String variable.

## Reference

Scott S, Jeanne B.: OCA/OCP Java SE 8 Programmer Practice Tests. Indiana, USA: 2017.