socrative

Answer Key
February 24, 2021

OCA Practice Questions Ch. 4 ('Copy')

1. What symbol is used for a varargs method parameter?

A ..

✓ B ...

C --

D ---

i Three dots (...) are the syntax for a method parameter of type varargs. It is treated like an array.

2.

```
public void toss (Frisbee... f) {
    Frisbee first = _____;
}
```

Fill in the blank in the following code to get the first element from the varargs parameter.

A f

✓ B f[0]

C f[1]

D None of the above

i Array indexes are zero based in Java. A varargs parameter is simply another way of passing in data to a method. From within the method, it is treated just like you had written Frisbee[] f as the method parameter. Therefore, the first element uses the 0th index, and Option B is correct.

3.

```
int[] lowercase = new int[0];
Integer[] uppercase = new Integer[0];
```

Which of the following are primitives?

A Only lowercase

B Only uppercase

C Bother lowercase and uppercase

✓ D Neither lowercase nor uppercase

i Trick question! While int is a primitive, all arrays are objects. One way to tell is that an array has a public instance variable called length. Another way is that you can assign it a variable of type Object. Therefore, Option D is correct.

4.

```
[]double lion;
double[] tiger;
double bear[];
```

How many of the following are legal declarations?

A None

B One

✓ C Two

D Three

i The array braces are allowed to appear before or after the variable name, making the tiger and bear declarations correct. The braces are not allowed to appear before the type making the lion declaration incorrect. Therefore, Option C is correct.

5.

```
public void printStormName(String... names) {
   System.out.println(Arrays.toString(names));
}
public void printStormNames(String[] names) {
   System.out.println(Arrays.toString(names));
}
```

Given the following two methods, which method call will not compile?

A printStormName("Arlene");

B printStormName(new String[] { "Bret" });

✓ C printStormNames("Cindy");

D printStormNames(new String[] { "Don" });

i From within a method, an array or varargs parameter is treated the same. However, there is a difference from the caller's point of view. A varargs parameter can receive either an array or individual values, making Options A and B compile. However, an array parameter can only take an array, which prevents Option C from compiling.

6. How do you determine the number of elements in an array?

✓ A buses.length

B buses.length()

C buses.size

D buses.size()

i Arrays use the length variable to determine the number of elements, making Option A correct. For an ArrayList, Option D would have been the answer.

7. Which of the following create an empty two-dimensional array with dimensions 2×2?

A int[][] blue = new int[2, 2];

B int[][] blue = new int[2], [2];

✓ C int[][] blue = new int[2][2];

D int[][] blue = new int[2 x 2];

i A two-dimensional array is declared by listing both sizes in separate pairs of braces. Option C correctly shows this syntax.

8.

```
String[] days = new String[] { "Sunday", "Monday", "Tuesday",
      "Wednesday", "Thursday", "Friday", "Saturday" };
for (int i = 0; i < days.length; i++)
      System.out.println(days[i]);
```

How many lines does the following code output?

A Six

✓ B Seven

C The code does not compile.

D The code compiles but throws an exception at runtime.

i There is nothing wrong with this code. It correctly creates a seven-element array. The loop starts with index 0 and ends with index 6. Each line is correctly output. Therefore, Option B is correct.

9. What are the names of the methods to do searching and sorting respectively on arrays?

A Arrays.binarySearch() and Arrays.linearSort()

✓ B Arrays.binarySearch() and Arrays.sort()

C Arrays.search() and Arrays.linearSort()

D Arrays.search() and Arrays.sort()

i Sorry. This is just something you have to memorize. The sort() and binarySearch() methods do sorting and searching, respectively.

10.

```
String[] nums = new String[] { "1", "9", "10" };
Arrays.sort(nums);
System.out.println(Arrays.toString(nums));
```

What does this code output?

A [1, 9, 10]

✓ B [1, 10, 9]

C [10, 1, 9]

D None of the above

i The elements of the array are of type String rather than int. Therefore, we use alphabetical order when sorting. The character 1 sorts before the character 9, alphabetically making Option A incorrect. Shorter strings sort before longer strings when all the other characters are the same, making Option B the answer.

11. Which of the following references the first and last element in a non-empty array?

A trains[0] and trains[trains.length]

✓ B trains[0] and trains[trains.length - 1]

C trains[1] and trains[trains.length]

D trains[1] and trains[trains.length - 1]

i Array indices start with 0, making Options C and D incorrect. The length attribute refers to the number of elements in an array. It is one past the last valid array index. Therefore, Option B is correct.

12.

```
String lion [] = new String[] {"lion"};
String tiger [] = new String[1] {"tiger"};
String bear [] = new String[] {};
String ohMy [] = new String[0] {};
```

How many of the following are legal declarations?

A None

B One

✓ C Two

D Three

i When using an array initializer, you are not allowed to specify the size separately. The size is inferred from the number of elements listed. Therefore, tiger and ohMy are incorrect. When you're not using an array initializer, the size is required. An empty array initializer is allowed. Option C is correct because lion and bear are legal.

13.

```
float[] lion = new float[];
float[] tiger = new float[1];
float[] bear = new[] float;
float[] ohMy = new[1] float;
```

How many of the following are legal declarations?

A None

✓ B One

C Two

D Three

i Since no elements are being provided when creating the arrays, a size is required. Therefore, lion and bear are incorrect. The braces containing the size are required to be after the type, making ohMy

incorrect. The only one that is correct is tiger, making the correct answer Option B.

14. Which statement most accurately represents the relationship between searching and sorting with respect to the Arrays class?

A If the array is not sorted, calling Arrays.binarySearch() will be accurate, but slower than if it were sorted.

B The array does not need to be sorted before calling Arrays.binarySearch() to get an accurate result.

✓ C The array must be sorted before calling Arrays.binarySearch() to get an accurate result.

D None of the above

i The binarySearch() method requires a sorted array in order to return a correct result. If the array is not sorted, the results of a binary search are undefined.

15. Which is not a true statement about an array?

✓ A An array expands automatically when it is full.

B An array is allowed to contain duplicate values.

C An array understands the concept of ordered elements.

D An array uses a zero index to reference the first element.

i An ArrayList expands automatically when it is full. An array does not, making Option A the answer. The other three statements are true of both an array and an ArrayList.

16.

```
String[][] matrix = new String[1][2];
matrix[0][0] = "Don't think you are, know you are.";       // m1
matrix[0][1] = "I'm trying to free your mind Neo";         // m2
matrix[1][0] = "Is all around you ";                       // m3
matrix[1][1] = "Why oh why didn't I take the BLUE pill?"; // m4
```

Which line of code causes an ArrayIndexOutOfBoundsException?

A m1

B m2

✓ C m3

D m4

i This code creates a two-dimensional array of size 1×2. Lines m1 and m2 assign values to both elements in the outer array. Line m3 attempts to reference the second element of the outer array. Since there is no such position, it throws an exception, and Option C is correct.

17.

```
String[] os = new String[] { "Mac", "Linux", "Windows" };
Arrays.sort(os);
System.out.println(Arrays.binarySearch(os, "Mac"));
```

What does the following output?

A 0

✓ B 1

C 2

D The output is not defined.

i The code sorts before calling binarySearch(), so it meets the precondition for that method. The target string of "Mac" is the second element in the sorted array. Since array indices begin with zero, the second position is index 1, and Option B is correct.

18.

```
char[][] ticTacToe = new char[3,3];                    // r1
ticTacToe[1][3] = 'X';                                 // r2
ticTacToe[2][2] = 'X';
ticTacToe[3][1] = 'X';
System.out.println(ticTacToe.length + " in a row!"); // r3
```

Which is the first line to prevent this code from compiling and running without error?

✓ A Line r1

B Line r2

C Line r3

D None of the above

i A multi-dimensional array is created with multiple sets of size parameters. The first line should be char[] ticTacToe = new char[3][3];. Therefore, Option A is the answer.

19.

```
Integer[] lotto = new Integer[4];
lotto[0] = new Integer(1_000_000);
lotto[1] = new Integer(999_999);
```

How many objects are created when running the following code?

A Two

✓ B Three

C Four

D Five

i The first line creates one object; the array itself. While there are four references to null in that array, none of those are objects. The second line creates one object and points one of the array references to it. So far there are two objects: the array itself and one object it is referencing. The third line does the same, bringing up the object count to three. Therefore, Option B is correct.

20.

```
[][] String alpha;
[] String beta;
String[][] gamma;
String[] delta[];
String epsilon[][];
```

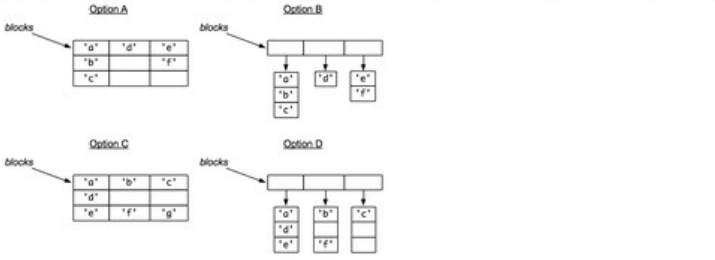
How many of the following are legal declarations?

- A Two
- ✓ B Three
- C Four
- D Five

i As with a one-dimensional array, the braces must be after the type, making alpha and beta illegal declarations. For a multi-dimensional array, the braces are allowed to be before and/or after the variable name. They do not need to be in the same place. Therefore, the remaining three are correct, and Option B is correct.

21.

```
char[][] blocks = new char[][] { { 'a', 'b', 'c' }, { 'd' }, { 'e', 'f' } };
```



Which of the options in the graphic best represent the blocks variable?

- A Option A
- ✓ B Option B
- C Option C
- D Option D

i Options A, C and D represent 3x3 2D arrays. Option B best represents the array in the code. It shows there are three different arrays of different lengths.

22.

```
public static void addStationName(String[] names) {
   names[names.length] = "Times Square";
}
```

What happens when calling the following method with a non-null and non-empty array?

- A It adds an element to the array the value of which is Times Square.
- B It replaces the last element in the array with the value Times Square.
- C It does not compile.
- ✓ D It throws an exception.

i names.length is the number of elements in the array. The last valid index in the array is one less than names.length. In Java, arrays do not resize automatically. Therefore, the code throws an ArrayIndexOutOfBoundsException.

23.

```
String[] days = new String[] { "Sunday", "Monday", "Tuesday",
      "Wednesday", "Thursday", "Friday", "Saturday" };
for (int i = 0; i < days.size(); i++)
      System.out.println(days[i]);
```

How many lines does the following code output?

- A Six
- B Seven
- ✓ C The code does not compile.
- D The code compiles but throws an exception at runtime.

i The code days.size() would be correct if this was an ArrayList. Since it is an array, days.length is the correct code. Therefore, the code does not compile, and Option C is the answer.

24.

```
boolean[][][] bools, moreBools;
```

How many dimensions does the array reference moreBools allow?

- A One dimension
- B Two dimensions
- ✓ C Three dimensions
- D None of the above

i Since the braces in the declaration are before the variable names, the variable type boolean[][][] applies to both variables. Therefore, both bools and moreBools can reference a 3D array.

25.

```
String[] strings = new String[2];
System.out.println(strings);
```

What is a possible output of the following code?

A [null, null]

B [,]

✓ C [Ljava.lang.String;@74a14482

D None of the above

i Calling toString() on an array doesn't output the contents of the array, making Option C correct. If you wanted Option A to be the answer, you'd have to call Arrays.toString(strings).

26.

```
char[][] ticTacToe = new char[3][3];                    // r1
ticTacToe[1][3] = 'X';                                  // r2
ticTacToe[2][2] = 'X';
ticTacToe[3][1] = 'X';
System.out.println(ticTacToe.length + " in a row!"); // r3
```

Which is the first line to prevent this code from compiling and running without error?

A Line r1

✓ B Line r2

C Line r3

D None of the above

i Arrays begin with an index of 0. This array is a 3×3 array. Therefore. only indexes 0, 1. and 2 are valid. Line r2 throws an ArrayIndexOutOfBoundsException. Therefore. Option B is correct.

27.

```
package duplicate;
public class Copier {
  public static void main(String... original) {
      String... copy = original;
      System.out.println(copy.length + " " + copy[0]);
   }
}
```

What is the result of running the following as java Copier?

A 0

B 0 followed by an exception

C 1 followed by an exception

✓ D The code does not compile.

i Three dots in a row is a varargs parameter. While varargs is used like an array from within the method, it can only be used as a method parameter. This syntax is not allowed for a variable, making Option D the answer.

28.

```
1:   package fun;
2:   public class Sudoku {
3:      static int[][] game = new int[6][6];
4:
5:      public static void main(String[] args) {
6:         game[3][3] = 6;
7:         Object[] obj = game;
8:         obj[3] = "X";
9:         System.out.println(game[3][3]);
10:     }
11:  }
```

What is the result of running the following program?

A X

B The code does not compile.

C The code compiles but throws a NullPointerException at runtime.

✓ D The code compiles but throws a different exception at runtime.

i Line 6 assigns an int to a cell in a 2D array. This is fine. Line 7 casts to a general Object[]. This is dangerous, but legal. Why is it dangerous, you ask? That brings us to line 8. The compiler can't protect us from assigning a String to the int[] because the reference is more generic. Therefore, line 8 throws an ArrayStoreException because the type is incorrect, and Option D is correct. You couldn't have assigned an int on line 8 either because obj[3] is really an int[] behind the scenes and not an int.

29.

```
String[] os = new String[] { "Mac", "Linux", "Windows" };
Arrays.sort(os);
System.out.println(Arrays.binarySearch(os, "RedHat"));
```

What does the following output?

A -1

B -2

✓ C -3

D The output is not defined.

i The code sorts before calling binarySearch, so it meets the precondition for that method. The target string of "RedHat" is not found in the sorted array. If it was found, it would be between the second and third element. The rule is to take the negative index of where it would be inserted and subtract 1. It would need to be inserted as the third element. Since indexes are zero based, this is index 2. We take the negative, which is −2, and subtract 1, giving −3. Therefore, Option C is correct.

30.

```
public class FirstName {
   public static void main(String... names) {
      System.out.println(names[0]);
   }
}
```

What is the output of the following when run as java FirstName Wolfie?

A FirstName

✓ B Wolfie

C The code throws an ArrayIndexOutOfBoundsException.

D The code throws a NullPointerException.

i Array indexes begin with zero. FirstName is the name of the class, not an argument. Therefore, the first argument is Wolfie, and Option B is correct.

31.

```
public class Count {
   public static void main(String target[]) {
      System.out.println(target.length);
   }
}
```

What is the output of the following when run as java Count 1 2?

A 0

B 1

✓ C 2

D The code does not compile.

i The name of the program is Count and there are two arguments. Therefore, the program outputs 2, and Option C is correct.

32. What is the output of the following when run as java EchoFirst seed flower?

A 0

✓ B 1

C The code does not compile.

D The code compiles but throws an exception at runtime.

```
package unix;
import java.util.*;
public class EchoFirst {

   public static void main(String[] args) {
      String one = args[0];
      Arrays.sort(args);
      int result = Arrays.binarySearch(args, one);
      System.out.println(result);
   }
}
```

i This class is called with two arguments. The first one (seed) is stored in the variable one. Then the array is sorted, meeting the precondition for binary search. Binary search returns 1 because seed is the second element in the sorted array, and Java uses zero-based indexes. Option B is correct.

33. Which of these four array declarations produces a different array than the others?

A int[][] nums = new int[2][1];

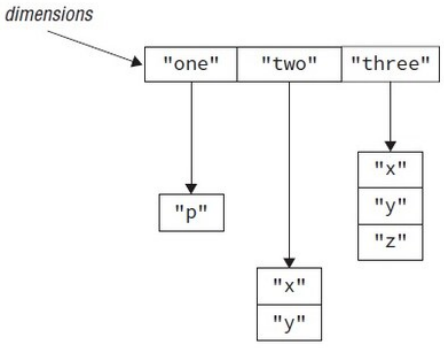
B int[] nums[] = new int[2][1];

C int[] nums[] = new int[][] { { 0 }, { 0 } };

✓ D int[] nums[] = new int[][] { { 0, 0 } };

i Options A and B show the braces can be before or after the variable name and produce the same array. Option C specifies the same array the long way with two arrays of length 1. Option D is the answer because it is different than the others. It instead specifies an array of length 1 where that element is of length 2.

34. How do you access the array element with the value of "z"?



A dimensions["three"][2]

B dimensions["three"][3]

✓ C dimensions[2][2]

D dimensions[3][3]

i Arrays are indexed using numbers, not strings, making Options A and B incorrect. Since array indexes are zero based, Option C is the answer.

35.

```
String[] days = new String[] { "Sunday", "Monday", "Tuesday",
      "Wednesday", "Thursday", "Friday", "Saturday" };
for (int i = 1; i <= days.length; i++)
      System.out.println(days[i]);
```

How many lines does the following code output?

A Six

B Seven

C The code does not compile.

✓ D The code compiles but throws an exception at runtime.

i In Java, arrays are indexed starting with 0. While it is unusual for the loop to start with 1, this does not cause an error. What does cause an error is the loop ending at data.length, because the <= operator is used instead of the < operator. The last loop index is 6, not 7. On the last iteration of the loop, the code throws an ArrayIndexOutOfBoundsException. Therefore, Option D is correct.

36.

```
public class FirstName {
   public static void main(String... names) {
      System.out.println(names[1]);
   }
}
```

What is the output of the following when run as java FirstName Wolfie?

A FirstName

B Wolfie

✓ C The code throws an ArrayIndexOutOfBoundsException.

D The code throws a NullPointerException.

i Array indexes begin with zero. FirstName is the name of the class, not an argument. The first and only argument is Wolfie. There is not a second argument, so Option C is correct.

37.

```
char[][] ticTacToe = new char[3][3];                  // r1
ticTacToe[0][0] = 'X';                                // r2
ticTacToe[1][1] = 'X';
ticTacToe[2][2] = 'X';
System.out.println(ticTacToe.length + " in a row!"); // r3
```

Which is the first line to prevent this code from compiling and running without error?

A Line r1

B Line r2

C Line r3

✓ D None of the above

i This code is correct. Line r1 correctly creates a 2D array. The next three lines correctly assign a value to an array element. Line r3 correctly outputs 3 in a row!

38.

```
public class Count {
   public static void main(String target[]) {
      System.out.println(target.length());
   }
}
```

What is the output of the following when run as java Count 1 2?

A 0

B 1

C 2

✓ D The code does not compile.

i Arrays expose a length variable. They do not have a length() method. Therefore, the code does not compile, and Option D is correct.

39.

```
boolean[][] bools[], moreBools;
```

How many dimensions does the array reference moreBools allow?

- A One dimension
- ✓ B Two dimensions
- C Three dimensions
- D None of the above

i This one is tricky since the array braces are split up. This means that bools is a 3D array reference. The braces both before and after the variable name count. For moreBools, it is only a 2D array reference because there are only two pairs of braces next to the type. In other words, boolean[][] applies to both variables. Then bools gets another dimension from the braces right after the variable name. However, moreBools stays at 2D, making Option B correct.

40.

```
package counting;
import java.util.*;
public class Binary {

   public static void main(String... args) {
      Arrays.sort(args);
      System.out.println(Arrays.toString(args));
   }
}
```

What is the result of the following when called as java counting.Binary?

- A null
- ✓ B []
- C The code does not compile.
- D The code compiles but throws an exception at runtime.

i Since no arguments are passed from the command line, this creates an empty array. Sorting an empty array is valid and results in an empty array. Therefore, Option B is correct.

41.

```
String[] os = new String[] { "Mac", "Linux", "Windows" };
System.out.println(Arrays.binarySearch(os, "Linux"));
```

What does the following output?

- A 0
- B 1
- C 2
- ✓ D The output is not defined.

i Java requires having a sorted array before calling binarySearch. Since the array is not sorted, the result is undefined, and Option D is correct. It may happen that you get 1 as the result, but this behavior is not guaranteed. You need to know for the exam that this is undefined even if you happen to get the “right” answer.

42.

```
1:   package fun;
2:   public class Sudoku {
3:      static int[][] game;
4:
5:      public static void main(String[] args) {
6:         game[3][3] = 6;
7:         Object[] obj = game;
8:         game[3][3] = "X";
9:         System.out.println(game[3][3]);
10:     }
11:  }
```

What is the result of running the following program?

A X

✓ B The code does not compile.

C The code compiles but throws a NullPointerException at runtime.

D The code compiles but throws a different exception at runtime.

i Line 8 attempts to store a String in an array meant for an int. Line 8 does not compile, and Option B is correct.

43.

```
String[][] listing = new String[][] { { "Book" }, { "Game", "29.99" } };
System.out.println(listing.length + " " + listing[0].length);
```

What is the output of the following?

✓ A 2 1

B 2 2

C The code does not compile.

D The code compiles but throws an exception at runtime.

i This array has two elements, making listing.length output 2. While each array element does not have the same size, this does not matter because we are only looking at the first element. The first element has one. This makes the answer Option A.

44.

```
public class FirstName {
   public static void main(String[] names) {
      System.out.println(names[0]);
   }
}
```

What is the output of the following when run as java FirstName?

A FirstName

B The code does not compile.

✓ C The code throws an ArrayIndexOutOfBoundsException.

D The code throws a NullPointerException.

i FirstName is the name of the class, not an argument. There are no other arguments, so names is an empty array. Therefore, Option C is correct.

45.

```
String[] days = new String[] { "Sunday", "Monday", "Tuesday",
      "Wednesday", "Thursday", "Friday", "Saturday" };
for (int i = 1; i < days.length; i++)
      System.out.println(days[i]);
```

How many lines does the following code output?

✓ **A** Six

B Seven

C The code does not compile.

D The code compiles but throws an exception at runtime.

i In Java, arrays are indexed starting with 0. While it is unusual for the loop to start with 1, this does not cause an error. It does cause the code to output six lines instead of seven since the loop doesn't cover the first array element. Therefore, Option A is correct.

46.

```
public class Count {
   public static void main(String target[]) {
      System.out.println(target.length);
   }
}
```

What is the output of the following when run as java Count "1 2"?

A 0

✓ **B** 1

C 2

D The code does not compile.

i The name of the program is Count, and there is only one argument because double quotes are used around the value. That argument is a String with three characters: 1, a space, and 2. Therefore, the program outputs 1, and Option B is correct.

47.

```
String[] os = new String[] { "Linux", "Mac", "Windows" };
System.out.println(Arrays.binarySearch(os, "Linux"));
```

What does the following output?

✓ **A** 0

B 1

C 2

D The output is not defined.

i Java requires having a sorted array before calling binarySearch(). You do not have to call Arrays.sort to perform the sort though. This array happens to already be sorted, so it meets the precondition. The target string of "Linux" is the first element in the array. Since Java uses zero-based indexing, the answer is Option A.

48. Which of the following statements are true?
I. You can always change a method signature from call(String[] arg) to call(String... arg) without causing a compiler error in the calling code.
II. You can always change a method signature from call(String... arg) to call(String[] arg) without causing a compiler error in the existing code.

✓ **A** I

B II

C Both I and II

D Neither I nor II

i From within a method, an array parameter and a varargs parameter are treated the same. From the caller, an array parameter is more restrictive. Both types can receive an array. However, only a varargs parameter is allowed to automatically turn individual parameters into an array. Therefore, statement I is correct and the answer is Option A.

49. Which of these four array references can point to an array that is different from the others?

A int[][][][] nums1a, nums1b;

✓ **B** int[][][] nums2a[], nums2b;

C int[][] nums3a[][], nums3b[][];

D int[] nums4a[][][], numbs4b[][][];

i All of the variables except nums2b point to a 4D array. Don't create a 4D array; it's confusing. The options show the braces can be before or after the variable in any combination. Option B is the answer because nums2b points to a 3D array. It only has three pairs of braces before the variable and none after. By comparison, nums2a has three pairs of braces before the variable and the fourth pair of braces after.

50.

```
package unix;
import java.util.*;
public class EchoFirst {

   public static void main(String[] args) {
      Arrays.sort(args);
      String result = Arrays.binarySearch(args, args[0]);
      System.out.println(result);
   }
}
```

What is the output of the following when run as java unix.EchoFirst seed flower?

A 0

B 1

✓ **C** The code does not compile.

D The code compiles but throws an exception at runtime.

i Binary search returns an int representing the index of a match or where a match would be. An int cannot be stored in a String variable. Therefore, the code does not compile and the answer is Option C.