Chapter 6 Arrays

1. See the section "Declaring and Creating Arrays."

2. You access an array using its index.

3. No memory is allocated when an array is declared. The memory is allocated when creating the array.

4. Indicate true or false for the following statements:
   1. Every element in an array has the same type.  
      Answer: True
   2. The array size is fixed after it is declared. 
      Answer: False
   3. The array size is fixed after it is created. 
      Answer: True
   4. The element in the array must be of primitive data type.  
      Answer: False

5. Which of the following statements are valid array declarations?

   int i = new int(30);  
   Answer: Invalid

   double d[] = new double[30];  
   Answer: Valid

   char[] r = new char(1..30);  
   Answer: Invalid

   int i[] = (3, 4, 3, 2);  
   Answer: Invalid

   float f[] = {2.3, 4.5, 5.6};  
   Answer: Valid

   char[] c = new char();  
   Answer: Invalid
6. The array index type is int and its lowest index is 0.

7. a [2]

8. A runtime exception occurs.

9. Line 3: the array declaration is wrong. It should be double[]. The array needs to be created before it's been used. E.g. new double[10]

   Line 5: The semicolon (;) at the end of the for loop heading should be removed.
   Line 5: r.length() should be r.length.
   Line 6: random should be random()
   Line 6: r(i) should be r[i].

10. System.arraycopy(source, 0, t, 0, source.length);


\[ \text{myList} \rightarrow \text{new int[10]} \]
\[ \text{myList} \rightarrow \text{new int[20]} \]

12. False. When an array is passed to a method, the reference value of the array is passed. No new array is created. Both argument and parameter point to the same array.

13.

numbers is 0 and numbers[0] is 3

14.
15. Only one variable-length parameter may be specified in a method and this parameter must be the last parameter. The method return type cannot be a variable-length parameter.
16. The last one
   ```java
   printMax(new int[]{1, 2, 3});
   ```
   is incorrect, because the array must of the `double[]` type.

17. Omitted

18. Omitted

19. Omitted

20. Simply change `(currentMax < list[j])` on Line 10 to `(currentMax > list[j])`

21. Simply change `list[k] > currentElement` on Line 9 to `list[k] < currentElement`

22. To apply `java.util.Arrays.binarySearch(array, key)`, the array must be sorted in increasing order.

23. You can sort an array of any primitive types except boolean. The sort method is `void`, so it does not return a new array.

24. Line 1: list is `{2, 4, 7, 10}`
   Line 2: list is `{7, 7, 7, 7}`
   Line 3: list is `{7, 8, 8, 7}`
   Line 4: list is `{7, 8, 8, 7}`

25. `int[ ][ ] m = new int[4][5];`

26. Yes. They are **ragged array**.

27. `array[0][1]` is 2.

28. `int[ ][ ] r = new int[2];`
   **Answer:** Invalid

   ```java
   int[ ] x = new int[];
   **Answer:** Invalid
   ```

   ```java
   int[ ][ ] y = new int[3][ ];
   **Answer:** Valid
   ```