1. What is the default value of Array for different data types?
Ans: The default of an array depends on the type of data stored in it. Different array types and the default value they will store is mentioned below:

Array Type(Which type of data it holds)	Default Value
int, byte, short	0
String	null
boolean	False
float	0.0f
double	0.0d
char	'\u0000'

2. Can you pass the negative number in Array size?
Ans: No, we cannot pass a negative number as Array size, since size of an array represents the number of elements in it, a negative number of elements in an array doesn't make any sense. Still, if we pass it, the program will get compiled successfully, but while executing it we will get a runtime exception of type **NegativeArraySizeException**.

3. Where does Array stored in JVM memory?
Ans: When an array is created, for example:

```
int arr[] = new int {1, 2, 3, 4, 5};
```

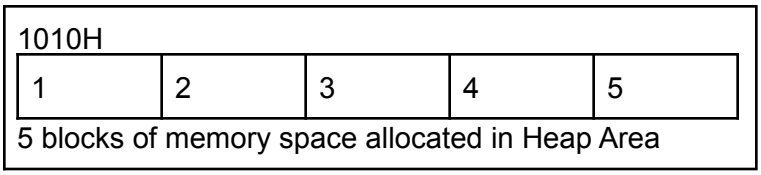
A block of memory is allocated for the array in the Heap Area of the JVM Memory and the variable name(here arr) alongwith the starting address of the block of memory is stored in the Stack Area of the JVM Memory.

Visualizing it:

Stack Area(The reference(or address) of the variable in Heap Area is stored in Stack Area)

arr	1010H

Heap Area



4. What are the disadvantages of Array?
Ans: The two major disadvantages of Array are:
i. Once an array is created, its size cannot be changed. It will remain fixed throughout the program/software/application. This is because during the time of array creation a fixed block of memory is created in the Heap Area which cannot be changed later on.
ii.An array in Java stores only homogeneous(similar) data elements. For example, an integer array can store only integer data and not String data.

5. What is an Anonymous Array in Java? Give an example.
Ans: An Anonymous array in Java is an array without a name. This type of array is created just for instance use.
Example:

```
import java.util.*;
//Program to multiply numbers in an array
class Demo{
   public int multiply(int nums[]){
       int result = 1;
       for(int n:nums){
           result = result * n;
       }
       return result;
   }
}

public class Program {
   public static void main(String args[]){
       Demo obj = new Demo();
       // Here, we are passing an array to multiply() method but not
       // creating a variable for the array like
       // int nums[] = {1, 2, 3, 4, 5}; and then passing it,
       // Instead, we are directly passing the value stored in the form
       // of an array without naming it
       int result = obj.multiply(new int[]{1, -1, -2, 3, 4});
       System.out.println(result);
   }
}
```

6. What are the different ways to traverse an Array in java?
Ans: We can traverse an array either using the for loop or the Enhanced for loop(for each loop).
Example:

```
int arr[] = {1, 2, 3, 4, 5, 6};
```

Using a for loop,

```
for(int i=0; i<=5;++i){
    arr[i] = 2 * arr[i];
}
```

Or,

```
for (int i=0; i< arr.length; ++i){
    arr[i] = 2 * arr[i];
}
```

Using the Enhanced for loop,

```
for(int n:arr){
arr[i] = 2 * arr[i];
}
```

Example:

```
int nums[][] = { {3, 6, 9},
                 {4, 7, 3},
                 {5, 2, 10}
               };
```

Using a for loop,

```
for(int i=0; i<3; ++i){
    for(int j=0; j<3; ++j){
        System.out.print(nums[i][j] + " ");
    }
System.out.println();
}
```

Using an enhanced for loop,

```
for(int i[]: nums){
    for(int j:i){
        System.out.print(j + " ");
    }
System.out.println();
}
```

7. What is the difference between length and length() method? Give an Example.
Ans: length is a property of the Array type class in Java and it is used to find out the size of an Array, i.e, number of elements in an array.
Example:

```
int arr[][] = {{3, 4, 6, 5, 7}, {10, 7}, {4,1}};
System.out.println(arr.length);    // this will print 3
System.out.println(arr[0].length); // this will print  5
```

length() is a method of the String class in Java and it is used to find out the length/size of the String, i.e, the number of characters the String is made up of.
Example:

```
String names[] = {"Rahul", “Rohit”, “Nagarjun”};
System.out.println(names.length);      // this will print 3
System.out.println(names[0].length()); // this will print 5
```