**1) What is ArrayStoreException in java? When you will get this exception?**

ArrayStoreException is a run time exception which occurs when you try to store non-compatible element in an array object. The type of the elements must be compatible with the type of array object. For example, you can store only string elements in an array of strings. if you try to insert integer element in an array of strings, you will get ArrayStoreException at run time.

public class MainClass

{

public static void main(String[] args)

{

Object[] stringArray = new String[5]; //No compile time error : String[] is auto-upcasted to Object[]

stringArray[1] = "JAVA";

stringArray[2] = 100; //No compile time error, but this statement will throw java.lang.ArrayStoreException at run time

//because we are inserting integer element into an array of strings

}

}

**2) Can you pass the negative number as an array size?**

No. You can’t pass the negative integer as an array size. If you pass, there will be no compile time error but you will get NegativeArraySizeException at run time.

public class MainClass

{

public static void main(String[] args)

{

int[] array = new int[-5]; //No compile time error

//but you will get java.lang.NegativeArraySizeException at run time

}

}

**3) Can you change the size of the array once you define it? OR Can you insert or delete the elements after creating an array?**

No. You can’t change the size of the array once you define it. You cannot insert or delete the elements after creating an array. Only you can do is change the value of the elements.

**4) What is an anonymous array? Give example?**

Anonymous array is an array without reference. For example,

public class MainClass

{

public static void main(String[] args)

{

//Creating anonymous arrays

System.out.println(new int[]{1, 2, 3, 4, 5}.length); //Output : 5

System.out.println(new int[]{21, 14, 65, 24, 21}[1]); //Output : 14

}

}

**5) What is the difference between int[] a and int a[] ?**

Both are the legal methods to declare the arrays in java.

**6) There are two array objects of int type. one is containing 100 elements and another one is containing 10 elements. Can you assign array of 100 elements to an array of 10 elements?**

Yes, you can assign array of 100 elements to an array of 10 elements provided they should be of same type. While assigning, compiler checks only type of the array not the size.

public class MainClass

{

public static void main(String[] args)

{

int[] a = new int[10];

int[] b = new int[100];

a = b; //Compiler checks only type, not the size

}

}

**7) “int a[] = new int[3]{1, 2, 3}” – is it a legal way of defining the arrays in java?**

No. You should not mention the size of the array when you are providing the array contents.

**8) What are the differences between Array and ArrayList in java?**

**Array** **ArrayList**

Arrays are of fixed length. ArrayList is of variable length.

You can’t change the size of the array once you create it. Size of the ArrayList grows and shrinks as you add or remove the elements.

Array does not support generics. ArrayList supports generics.

You can use arrays to store both primitive types as well as reference types. You can store only reference types in an ArrayList.

**9) What are the different ways of copying an array into another array?**

There are four methods available in java to copy an array.

1) Using **for loop**

2) Using **Arrays.copyOf()** method

3) Using **System.arraycopy()** method

4) Using **clone()** method

Click here to see these methods in detail.

**10) What are jagged arrays in java? Give example?**

Jagged arrays in java are the arrays containing arrays of different length. Jagged arrays are also multidimensional arrays. They are also called as ragged arrays.

**11) How do you check the equality of two arrays in java? OR How do you compare the two arrays in java?**

You can use **Arrays.equals()** method to compare one dimensional arrays and to compare multidimensional arrays, use **Arrays.deepEquals()** method.

**12) What is ArrayIndexOutOfBoundsException in java? When it occurs?**

ArrayIndexOutOfBoundsException is a run time exception which occurs when your program tries to access invalid index of an array i.e negative index or index higher than the size of the array.

**13) How do you sort the array elements?**

You can sort the array elements using **Arrays.sort()** method. This method internally uses quick sort algorithm to sort the array elements.

import java.util.Arrays;

public class MainClass

{

public static void main(String[] args)

{

int[] a = new int[]{45, 12, 78, 34, 89, 21};

Arrays.sort(a);

System.out.println(Arrays.toString(a));

//Output : [12, 21, 34, 45, 78, 89]

}

}

**14) How do you find the intersection of two arrays in java?**

[Answer]

**15) What are the different ways of declaring multidimensional arrays in java?**

The following code snippet shows different ways of declaring 2D, 3D and 4D arrays.

**//2D Arrays**

int[][] twoDArray1;

int twoDArray2[][];

int[] twoDArray3[];

**//3D Arrays**

int[][][] threeDArray1;

int threeDArray2[][][];

int[] threeDArray3[][];

int[][] threeDArray4[];

**//4D Arrays**

int[][][][] fourDArray1;

int fourDArray2[][][][];

int[] fourDArray3[][][];

int[][] fourDArray4[][];

int[][][] fourDArray5[];

**16) While creating the multidimensional arrays, can you specify an array dimension after an empty dimension?**

No. You can not specify an array dimension after an empty dimension while creating multidimensional arrays. It gives compile time error.

int[][][] a = new int[][5][]; //Compile time error

int[][][] b = new int[5][][5]; //Compile time error

int[][][] c = new int[][5][5]; //Compile time error

**17) How do you search an array for a specific element?**

You can search an array to check whether it contains the given element or not using **Arrays.binarySearch()** method. This method internally uses binary search algorithm to search for an element in an array.

**18) What value does array elements get, if they are not initialized?**

They get default values.

**19) How do you find duplicate elements in an array?**

[Answer]

**20) What are the different ways to iterate over an array in java?**

1) Using normal for loop

public class MainClass

{

public static void main(String[] args)

{

int[] a = new int[]{45, 12, 78, 34, 89, 21};

//Iterating over an array using normal for loop

for (int i = 0; i &lt; a.length; i++)

{

System.out.println(a[i]);

}

}

}

2) Using extended for loop

public class MainClass

{

public static void main(String[] args)

{

int[] a = new int[]{45, 12, 78, 34, 89, 21};

//Iterating over an array using extended for loop

for (int i : a)

{

System.out.println(i);

}

}

}

**21) How do you find second largest element in an array of integers?**

[Answer]

**22) How do you find all pairs of elements in an array whose sum is equal to a given number?**

[Answer]

**23) How do you separate zeros from non-zeros in an integer array?**

[Answer]

**24) How do you find continuous sub array whose sum is equal to a given number?**

[Answer]

**25) What are the drawbacks of the arrays in java?**

The main drawback of the arrays is that arrays are of fixed size. You can’t change the size of the array once you create it. Therefore, you must know how many elements you want in an array before creating it. You can’t insert or delete the elements once you create an array. Only you can do is change the value of the elements.

**The followings are 10 interesting observations about arrays in java. You may be asked in the interviews or Java certification exams about these observations.**

1) The size of an array can not be negative. If you give size of an array as negative, you don’t get any errors while compliing. But, you will get NegativeArraySizeException at run time.

public class ArraysInJava

{

public static void main(String[] args)

{

int[] i = new int[-5]; //No Compile Time Error

//You will get java.lang.NegativeArraySizeException at run time

}

}

2) The size of an array must be an integer or an expression which results an integer. Auto-widening is also allowed.

public class ArraysInJava

{

public static void main(String[] args)

{

int[] i = new int[10+5]; //Array size can be an expression resulting an integer

int[] i1 = new int[(byte)10]; //byte is auto-widened to int

int i3 = new int[10.25]; //Compile Time Error : size can not be double

}

}

3) Declaration and instantiating of an array strictly must be of same type. No auto-widening, auto-boxing and auto-unboxing is allowed. But only auto-upcasting is allowed.

public class ArraysInJava

{

public static void main(String[] args)

{

Integer[] I = new int[5]; //Compile Time Error : Auto-Boxing not allowed

int[] i = new Integer[10]; //Compile Time Error : Auto-UnBoxing not allowed

long[] l = new byte[10]; //Compile Time Error : Auto-widening not allowed

Object[] o = new String[10]; //No Compile Time Error : Auto-Upcasting is allowed, String[] is upcasted to Object[]

}

}

4) The type of elements of an array must be compatible with type of the array object. If you try to store non-compatible element in an array object, you will get ArrayStoreException at run time.

public class ArraysInJava

{

public static void main(String[] args)

{

Object[] o = new String[10]; //No Compile Time Error : String[] is auto-upcasted to Object[]

//i.e array object of strings can be referred by array reference variable of Object type

o[2] = "java";

o[5] = 20; //No Compile time error,

//but you will get java.lang.ArrayStoreException at run time.

}

}

5) If you are supplying the contents to the array without new operator, then it should be at the time of declaration only. Not at any other places.

public class ArraysInJava

{

public static void main(String[] args)

{

int[] i = {1, 2, 3, 4}; //This is the correct way

i = {1, 2, 3 , 4}; //Compile time error

i = new int[]{1, 2, 3, 4}; //This is also correct way

}

}

6) Another way of declaring multi dimensional arrays.

public class ArraysInJava

{

public static void main(String[] args)

{

int[][] twoDArray; //Normal way of declaring two-dimensional array

int[] TwoDArray []; //Another way of declaring two-dimensional array

int[][][] threeDArray; //Normal way of declaring three-dimensional array

int[] ThreeDArray [][]; //This is also legal

}

}

7) While creating multi dimensional arrays, you can not specify an array dimension after an empty dimension.

public class ArraysInJava

{

public static void main(String[] args)

{

int[][][] threeDArray = new int[10][][10]; //Compile Time Error

int[][][] threeDArray1 = new int[][10][]; //Compile Time Error

int[][][] threeDArray2 = new int[][][10]; //Compile Time Error

}

}

8) You can create an anonymous array i.e an array without reference.

public class ArraysInJava

{

public static void main(String[] args)

{

//Creating anonymous array

System.out.println(new int[]{1, 2, 3}.length); //Output : 3

System.out.println(new int[]{47, 21, 58, 98}[1]); //Output : 21

}

}

9) While assigning one array reference variable to another, compiler checks only type of the array not the size.

public class ArraysInJava

{

public static void main(String[] args)

{

int[] a = new int[10];

int[] b = new int[100];

double[] c = new double[20];

a = b;

b = c; //Compile Time Error : can not convert from double[] to int[]

}

}

10) The size of an array can not be changed once you define it. You can not insert or delete array elements after creating an array. Only you can change is the value of the elements. This is the main drawback of arrays.