Practical 5 : Questions

Q1. How to create an array in Java?

Data\_type array\_name [] = new data\_type [size];

OR

Data\_type [] array\_name = new data\_type [size];

Example : int a[] = new int [5];

Q2. How to create String type of array in Java?

String array\_name[] = new String [size];

Q3. How to initialize array elements in JAVA?

Int arr[] = {10,20,30}

OR

Int arr[] = new int [3];

arr[0] = 10;

arr[1] = 20;

arr[2] = 30;

OR

Import java.util.Scanner;

Scanner sc = new Scanner (System.in);

Int arr[] = new int [3];

for (int I = 0; i<3; i++){

arr[i] = sc.nextInt();

}

Q4. How to display array elements in JAVA?

For (int i = 0; i<array\_name.length; i++){

System.out.print(arr\_name[i] + “ ,”);

}

OR

For (int I : arr){

System.out.println(I);

}

Q5. How to sort array elements in JAVA? OR

What are the inbuild methods to sort out the array elements?

Using user-defined functions : selection sort

Insertion sort

Bubble sort

Merge sort

Quick sort

Count sort

Using inbuilt functions:

For ascending order:

import java.util.Arrays;

Arrays.sort(arr\_name);

For descending order:

Import java.util.Arrays;

Import java.util.Collections;

Arrays.sort(a,Collections.reverseOrder());

// reverseOrder function does not work for inbuilt data types

Graphical user interface, text

Description automatically generated with medium confidenceQ6. How to create an array of different types of elements?

Q7. How to create dynamic type of array?

We make a class for the array and there we write methods for add element at index I, grow size, etc

Q8. What is the difference between int and Integer type of data?

In Java, **int is a primitive data type while Integer is a Wrapper class**.

* int, being a primitive data type has got less flexibility. We can only store the binary value of an integer in it.
* Since Integer is a [wrapper class](https://www.geeksforgeeks.org/wrapper-classes-java/) for int data type, it gives us more flexibility in storing, converting and manipulating an int data.
* Graphical user interface, text, application

  Description automatically generatedInteger is a class and thus it can call various in-built methods defined in the class. Variables of type Integer store references to Integer objects, just as with any other reference (object) type.

Q9. What is the difference between String s and String s = new String[]?

String s = new String(“A”); explicitly creates a new and referentially distinct instance of a String object.

String s = “A”; will reuse an instance from the **string constant pool** if one is available.

—————————————————————

String s1 = "quora";

String s2 = "quora";

System.out.println(s1 == s2); // true

s2 = new String("quora");

System.out.println(s1 == s2); // false

System.out.println(s1.equals(s2)); // true

—————————————————————-

In simpler words, when **String s1 = “quora”;** is executed, a new memory box is created with value as “quora” and name as “s1”. Now when **String s2 = “quora”;** takes place, then the variable “s2” points to the memory reference of s1 whose value is “quora”.

That is, if the value is same (as in this case, value is “quora”) then JVM does not create a new memory reference. Instead it points to the already made reference and thus memory management is practiced. But now if you were to create **String s3 = “abcd”;** Now the JVM will check the if any memory box is present in the String pool where “abcd” value is already present. If yes, it will point variable ‘s3’ to this box otherwise it will create a new one.

On the other hand, **String s4 = new String(“quora”);** will explicitly create a new memory reference in the memory.

\*Remember, the “new” keyword will always create something new in the heap memory.