Arrays in Java

Assignment Questions:

1. What is the default value of Array for different data types?

Ans → The default value of an array in Java depends on the data type of the array. Here are the default values for the common data types:

* For arrays of primitive data types (such as int, double, boolean, etc.), the default value of each element is 0, false, or the equivalent "empty" value for the data type.
* For arrays of reference data types (such as String, Object, or any other class), the default value of each element is null.

1. Can you pass the negative number in Array size?

Ans → No, we cannot create an array with a negative size in Java. If we try to create an array with a negative size, you will get a NegativeArraySizeException.

1. Where does Array stored in JVM memory?

Ans →In Java, arrays are stored in the heap memory of the Java Virtual Machine (JVM). When you create an array using the new operator, the JVM allocates a contiguous block of memory on the heap to store the elements of the array.

1. What are the disadvantage of Array?

Ans → i. Fixed-size: The size of an array is fixed and cannot be changed dynamically at runtime. Once an array is created, its size cannot be changed. This can be limiting in cases where you need to add or remove elements from the array.

ii. Inefficient insertion and deletion: Inserting or deleting an element in an array requires shifting all the subsequent elements. This can be a slow operation, especially for large arrays. If you need to insert or delete elements frequently, using an array may not be the best choice.

iii. Wasted memory: If you allocate an array that is larger than you need, you may waste memory. This can be a problem if you are working with very large arrays or if memory is limited.

iv. Homogeneous data: An array can only store elements of a single data type. If you need to store elements of different data types, you may need to use an array of objects.

1. What is an Anonymous Array in java? Give an example.

Ans → In Java, an anonymous array is an array that is created without assigning it to a named variable. Instead, the array is created and initialized in a single line of code, and its reference is used immediately as an argument to a method or as a value in an expression.

System.out.println("The sum of the first 3 numbers is " +

Arrays.stream(new int[] { 1, 2, 3 }).sum());

1. What are the different ways to traverse an Array in JAVA?

Ans → Method 1: Using for loop: You can use a for loop to iterate over the elements of an array. The loop will start from the first element and continue until the last element. Here's an example:

int[] numbers = {1, 2, 3, 4, 5};

for (int i = 0; i < numbers.length; i++) {

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

}

Method 2: Using for-each loop: this is a simpler way to traverse an array that was introduced in Java 5. It iterates over each element in the array without the need for an index variable. Here's an example:

int[] myArray = {1, 2, 3, 4, 5};

for (int element : myArray) {

System.out.println(element);

}

Method 3: Using while loop: You can also use a while loop to traverse an array by initializing an index variable outside the loop and incrementing it inside the loop until it reaches the end of the array. Here's an example:

int[] myArray = {1, 2, 3, 4, 5};

int i = 0;

while (i < myArray.length) {

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

i++;

}

Method 4: Using do-while loop: This is a similar approach to the while loop, but the loop condition is checked after the loop body is executed. Here's an example:

int[] myArray = {1, 2, 3, 4, 5};

int i = 0;

do {

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

i++;

} while (i < myArray.length);

1. What is the difference between length and length() method Give an example?

Ans → The key difference between length and length() is that length is a property of an array that returns an integer value, whereas length() is a method of the String class that returns the number of characters in a string.

// length method

int[] myArray = {1, 2, 3, 4, 5};

int length = myArray.length;

System.out.println(length); // Output: 5

//length() method

String myString = "Hello, World!";

int length = myString.length();

System.out.println(length); // Output: 13