Assignment – 10th (Arrays in Java)

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

Ans: - The default value of an array in different data types is as follows:

- For numeric data types (such as ***`int`***, ***`float`***, ***`double`***, etc.), the default value is ***`0`*** for all elements of the array.
- For ***`char`*** data type, the default value is the null character ***'\0'*** for all elements of the array.
- For ***`boolean`*** data type, the default value is ***`false`*** for all elements of the array.
- For reference data types (such as objects, arrays, etc.), the default value is ***`null`*** for all elements of the array.

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

Ans: - No, you cannot pass a negative number as the size of an array in Java. Java arrays have a non-negative integer size, and attempting to create an array with a negative size will result in a NegativeArraySizeException being thrown. Example:

```
int arraySize = -5;
int arrayName[] = new int[arraySize];
```

In this example, the size of the array is set to ***`-5`***, which is a negative value. When the ***`new`*** keyword is used to allocate memory for the array, a ***`NegativeArraySizeException`*** is thrown, indicating that the size of the array must be a non-negative integer.

3. Where does Array stored in JVM memory?

Ans: - Arrays in Java are stored in the heap, which is a portion of the JVM's memory space where objects are stored. When you create an array in Java, the JVM will allocate a block of memory in the heap to store the elements of the array. This block of memory will have a size that is determined by the size of the array and the size of the elements it contains. This means that arrays in Java are dynamic data structures that can grow and shrink as needed during the execution of a program.

For example, an array of ***`int`*** values will require a different amount of memory than an array of ***`String`*** objects, because ***`int`*** values are smaller than ***`String`*** objects.

4. What are the disadvantages of Array?

Ans: - Disadvantages of Array in Java:

- The size of the array cannot be increased or decreased once it is declared—arrays have a fixed size.
- You cannot insert a new element at the middle of the array. In the same way you cannot delete elements from the middle of the array. You can only insert/delete from the end of the array.
- Java cannot store heterogeneous data. It can only store a single type of primitives.

5. What is an Anonymous Array in Java, Give an example?

Ans: - An anonymous array in Java is an array that is created without giving it a name. Anonymous arrays are typically used when you need to pass an array as an argument to a method or when you need to create an array for a one-time use.

Here's an example of an anonymous array in Java:

```
class Calc{

        int add(int nums[]){
                int total = 0;
                for(int n : nums){
                        total = total+n;
                }
        return total;
        }

}

public class Main {
        public static void main(String[] args) {
                //creating an anonymous array and passing it to the sum metod
                Calc obj = new Calc();
                int total = obj.add(new int[]{1,2,3,4,5});
                System.out.println("Sum of elements: " + total);
        }
}
```

In this example, we create an anonymous array new ***`int[]{1, 2, 3, 4, 5}`*** and pass it to the ***`sum`*** method. The ***`add`*** method calculates the sum of the elements in the array and returns the result.

6. What are the different ways to traverse an Array in Java?

Ans: - There are several ways to traverse an array in Java:

- **For loop: -** This is the most common way to traverse an array in Java. You can use a for loop to iterate through each element of the array and access its value. For example:

```
int nums[] = {1, 2, 3, 4, 5};
for(int i=0; i<num.length; i++){
        System.out.println(nums[i]);
}
```

- **Enhanced For loop: -** This is also known as a for-each loop, and it is a simpler way to traverse an array in Java. The enhanced for loop allows you to iterate through each element of the array without having to access the array's indices. For example:

```
int nums[] = {1, 2, 3, 4, 5};
for(int n : nums){
        System.out.println(n);
}
```

7. What is the difference between length and length() method, Give an Example?

Ans: - In Java, ***"length"*** is a property of arrays, while ***"length()"*** is a method of the String class.

Here's an example to help illustrate the difference:

```
int nums[] = new int[5];
int arrayLength = nums.length;
```

In this example, "nums" is an array of integers with a length of 5, which is stored in the ***`arrayLength`*** variable.

```
String name = "Rahul Kumar";
int stringLength = name.length();
```

In this example, ***`name`*** is a string with a length of 4, which is determined by calling the ***`length()`*** method and stored in the ***`stringLength`*** variable.

So, the main difference between **"length"** and **"length()"** is that **"length"** is a property of arrays, while **"length()"** is a method of the String class.