Day-13 Assignment Solution

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

In Java, when you declare an array but do not initialize it, the elements of the array get default values based on their data types:

- For numeric data types (byte, short, int, long, float, double): 0
- For the char data type: '\u0000' (null character)
- For boolean data type: false
- For reference data types (objects): null

These default values are assigned automatically when you create an array but do not explicitly initialize its elements.

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

No, in Java, you cannot pass a negative number as the size of an array. The size of an array must be a non-negative integer. Attempting to create an array with a negative size will result in a runtime exception called `NegativeArraySizeException`.

3. Where does Array store in JVM memory?

In Java, arrays are stored in the Java Virtual Machine (JVM) heap memory. The heap is a region of memory where Java objects, including arrays, are dynamically allocated. When you create an array in Java, memory is allocated from the heap to store the elements of the array. The array itself is an object in Java, so it is stored on the heap like any other object.

4. What are the disadvantages of Array?

Arrays in Java have several disadvantages:

1. Fixed Size: Arrays have a fixed size, meaning you must specify the size of the array when you declare it, and this size cannot be changed once the array is created.
2. Lack of Dynamic Resizing: Unlike some other data structures like ArrayList, arrays cannot dynamically resize themselves. If you need to add or remove elements from an array, you must create a new array with the desired size and copy the elements over.
3. Inefficient Insertion and Deletion: Inserting or deleting elements in the middle of an array requires shifting all subsequent elements, which can be inefficient, especially for large arrays.
4. Not Suitable for Heterogeneous Data: Arrays are homogeneous data structures, meaning they can only store elements of the same data type. It's not convenient to store elements of different types in the same array.
5. Memory Wastage: If you allocate more space than needed for the array, it can lead to memory wastage.
6. Lack of Built-in Methods: Arrays in Java lack built-in methods for common operations like searching, sorting, and inserting elements at specific positions.
7. No Built-in Bounds Checking: Arrays in Java do not perform bounds checking by default, so accessing elements beyond the array bounds can result in runtime errors.

5. What is an Anonymous Array in java?

In Java, an anonymous array is an array that is created without explicitly specifying its type and size using the "new" keyword. Instead, it is created and initialized in a single line of code. Anonymous arrays are typically used in situations where an array is needed for a short duration and it's not necessary to assign it to a variable.

```
class Test{
    public int add(int nums[]){
        int result = 0;
        for(int n : nums){
            result += n;
        }
        return result;
    }
}
public class AnnonymousArray {
    public static void main(String[] args) {
        Test obj = new Test();
        int result = obj.add(new int[]{1,2,3,4,5});//Anonymous Array
        System.out.println(result);
    }
}
```

Anonymous arrays are useful in scenarios where you need to pass an array as an argument to a method or as part of an array initialization, and you don't need to reuse the array elsewhere in your code. They provide a concise way to create and use arrays without declaring a separate variable.

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

In Java, there are multiple ways to traverse or iterate over an array, each with its own advantages and use cases:

For Loop: A traditional for loop can be used to iterate over each element of the array sequentially. This method provides full control over the loop and is suitable for iterating over arrays when the index of each element is needed.

```
int[] array = {1, 2, 3, 4, 5};
for (int i = 0; i < array.length; i++) {
    // Access array elements using index i
    System.out.println(array[i]);
}
```

Enhanced For Loop (Foreach): Introduced in Java 5, the enhanced for loop simplifies array traversal by automatically iterating over each element without needing an explicit index variable. This method is concise and easy to read but does not provide access to the index of each element.

```
int[] array = {1, 2, 3, 4, 5};
for (int num : array) {
    // Access array elements directly
    System.out.println(num);
}
```

While Loop: You can use a while loop along with an index variable to traverse the array. This method offers flexibility in loop termination conditions and is suitable for cases where you need to control the iteration flow manually.

```
int[] array = {1, 2, 3, 4, 5};
int i = 0;
while (i < array.length) {
    // Access array elements using index i
    System.out.println(array[i]);
    i++;
}
```

7. What is different between length and length() method and Give an Examples?

length: 'length' is a property of arrays. It returns the number of elements in the array. It's used directly on arrays without parentheses.

length(): 'length()' is a method of the String class (and also of arrays in some languages like JavaScript). It returns the number of characters in the string. It's used as a method call on a String object.

```
public class Length {
    public static void main(String[] args) {
        String name = "Md Munna";
        System.out.println(name.length());
        int n[] = {1,2,3,5};
        System.out.println(n.length);
    }
}
```