In Java, the **Arrays** class provides several static methods for working with arrays. These methods are often used for tasks like sorting, searching, and manipulating arrays. Below is a list of some common methods provided by the **Arrays** class:

1. **asList(T... a)**: Converts an array to a fixed-size list.
2. **binarySearch(T[] a, T key)**: Searches for a specified element in a sorted array using binary search.
3. **binarySearch(T[] a, int fromIndex, int toIndex, T key)**: Searches for a specified element in a sorted subarray using binary search.
4. **copyOf(T[] original, int newLength)**: Copies the specified array, truncating or padding with zeros to obtain the desired length.
5. **copyOfRange(T[] original, int from, int to)**: Copies the specified range of elements from the source array to a new array.
6. **equals(T[] a, T[] a2)**: Compares two arrays for equality.
7. **fill(T[] a, T val)**: Assigns the specified value to every element in the array.
8. **fill(T[] a, int fromIndex, int toIndex, T val)**: Assigns the specified value to a range of elements in the array.
9. **hashCode(T[] a)**: Computes a hash code for the array.
10. **sort(T[] a)**: Sorts the array in ascending order.
11. **sort(T[] a, int fromIndex, int toIndex)**: Sorts a range of elements in the array in ascending order.
12. **toString(T[] a)**: Returns a string representation of the array.
13. **deepEquals(Object[] a1, Object[] a2)**: Performs a deep comparison of two arrays to determine if they are equal.
14. **deepHashCode(Object[] a)**: Computes a hash code for a deep array comparison.
15. **deepToString(Object[] a)**: Returns a string representation of a deep array.

Note that these methods are part of the **java.util.Arrays** class and are often used with arrays of primitive types (e.g., **int[]**, **double[]**) as well as arrays of objects (e.g., **String[]**, **Object[]**). These methods provide convenient utilities for working with arrays in Java.

Top of Form