**Problem: Implement 3 methods that can be used to return an array that has no duplicates**.

**Solution1:**

Implementation using Hash Set Data Structure.

Advantage - Use of standard java data structure. Reduced development effort by using core collection classes rather than implementing our own collection classes.

Disadvantage - Multiple looping of input dataset. Once to add to set and then to convert back into int array as return value.

It is possible to clean the code with use of third party libraries like Apache ArrayUtils to convert set back into int array, but the looping is still unavoidable.

Input Integer Array has 78 elements

1 2 34 34 25 1 45 3 26 85 4 34 86 25 43 2 1 10000 11 16 19 1 18 4 9 3 20 17 8 15 6 2 5 10 14 12 13 7 8 9 1 2 15 12 18 10 14 20 17 16 3 6 19 13 5 11 4 7 19 16 5 9 12 3 20 7 15 17 10 6 1 8 18 4 14 13 2 11

Output from Set Implementation has 28 elements and took 1 milliseconds

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 10000 16 17 18 19 20 85 86 25 26 34 43 45

**Solution 2:**

Removes Duplicates using additional primitive int array.

Advantage - Use of primitive array.

Disadvantage - Multiple looping of array. The total size required would also be large as second array size is dependent on the largest value in the array. Works only with values greater than zero. We need to know the size of the array in advance. Since array is of fixed size, if we allocate more memory than requirement then the memory space will be wasted.

Remove Duplicates with Array Implementation took 1 milliseconds

Input Integer Array has 78 elements

1 2 34 34 25 1 45 3 26 85 4 34 86 25 43 2 1 10000 11 16 19 1 18 4 9 3 20 17 8 15 6 2 5 10 14 12 13 7 8 9 1 2 15 12 18 10 14 20 17 16 3 6 19 13 5 11 4 7 19 16 5 9 12 3 20 7 15 17 10 6 1 8 18 4 14 13 2 11

Output from Array Implementation has 28 elements and took 1 milliseconds

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 25 26 34 43 45 85 86 10000

**Solution 3**:

Removes duplicates with the use of Array List

Advantage - Use of standard Java Data Structure. Array List maintains the order of input array.

Disadvantage - Multiple looping of input dataset. Once to add to list and then to convert back into int array as return value.

It is possible to clean the code with use of third party libraries like Apache ArrayUtils to convert List back into int array.

Input Integer Array has 78 elements

1 2 34 34 25 1 45 3 26 85 4 34 86 25 43 2 1 10000 11 16 19 1 18 4 9 3 20 17 8 15 6 2 5 10 14 12 13 7 8 9 1 2 15 12 18 10 14 20 17 16 3 6 19 13 5 11 4 7 19 16 5 9 12 3 20 7 15 17 10 6 1 8 18 4 14 13 2 11

Output from List Implementation has 28 elements and took 0 milliseconds

1 2 34 25 45 3 26 85 4 86 43 10000 11 16 19 18 9 20 17 8 15 6 5 10 14 12 13 7