Arrays

Array is a collection of similar types of data.

```
int []marks = new int[5];
```
(5 → size)

or

```
int[] marks = new int[5];
```

or

```
int marks[] = new int[5];
```

Array index starts from zero

Accessing array elements:

```
marks arr[0] = 100;
      arr[1] = 70;
```

Ways of Declaration

1) `int[] arr;` → Declaration
   `arr = new int[5];` → Memory Allocation

1) ~~int[] marks = new int[5]~~

2) int arr[] = new int [5] → Declaration + Memory Allocation

3) int [] arr = {100, 71, 80, 70} → Declare + Initialize

Array starts from 0 & goes to n-1 where n is size.

Array length

This gives the length of array

```
arr.length
```

Displaying an array

```
for (i=0; i<marks.length; i++)
{
    ~~Sout(marks)~~
    Sout(arr[i]);
}
```

For each loop

→ For each loop is an enhanced version of for loop.
→ It travels each element one by one
→ You cannot skip ~~the~~ any element in a for loop & it's also not possible to traverse elements in reverse order.

→ It increases readability of the code.

Syntax:

```
for(int element : Arr)
{
    Sout(element); // Prints all elements
}
```

Multi-dimensional array

2-D array

```
int [][] arr = new int [2][3];
```

↳ 2D array of 2 rows & 3 columns

2-D Array visualization

	Col 1 [0]	Col 2 [1]	Col 3 [2]
[0] Row 1	(0,0)	(0,1)	(0,2)
[1] Row 2	(1,0)	(1,1)	(1,2)

3 D array of strings

```
String [][][] arr = new String [2][3][4]
```

Practice Problems

Q1 Write a program to find out whether a given integer is present in the array or not.

Q2 Write a Java program to reverse the ~~a~~ an array

Q3 ~~Wap~~ Find out whether the array is sorted or not

```
package com.company;

public class arrays {
    public static void main (String[] args)
    {
    // Problem 1
    float marks[] = {45.7f, 67.8f, 63.4f, 99.2f,
                                        100.0f};
    float num = 45.57f;

    boolean isInArray = false;

    for (float element : marks)
        {
            if (num == element)
            {
                isInArray = true;
                break;
            }
        }
    if (isInArray) {
        Sout ("The value is present int the
                                        array");
        }
    else {
        Sout ("The value is not present in the
                                        array");
```

// Problem 2

To find the G.I. of DRUPSOM

```
int [] arr = {1, 21, 3, 4, 5, 34, 67};
int l = arr.length;
int n = Math.floor Div(l, 2);
int temp;

for(i=0; i<n; i++)
  {
   temp = arr[i];
   arr[i] = arr[l-i-1];
   arr[l-i-1] = temp;
  }

  for(int element : arr)
  {
   Sout(element + " ");
  }
```

// Problem 3

```
int boolean isSorted = true;
int [] arr = {1,12,3,4,5,34,67};

for(int i=0; i<arr.length-1; i++)
  {
    if(arr[i] > arr[i+1])
    {
      isSorted = false;
      break;
    }
  }
```

```
if (isSorted){
    Sout ("Array is sorted");
}
else
{
Sout ("Array isn't sorted");
}
}
}
```