Java Array:

Normally, array is a collection of similar type of elements that have contiguous memory location. Contiguous means without any gap.

For example: Suppose a bats man played 6 balls and over ended. This is contagious but the same batsmen will play after few ball of gaps that is continuous.

Java array is an object that contains elements of similar data type. It is a data structure where we store similar elements. We can store only fixed set of elements in a Java array.

Array in Java is index based, first element of the array is stored at 0 index. Array is a reference variable type.

Once an Array has got a memory, it can not be increased and It can not be decreased. Before Java Arrays were getting the memory statically, That’s why it were required to allocate the size of array while defining. But now in Java we can declare, instantiate and Initialize separately.

Advantage of Java Array

Code Optimization: It makes the code optimized, we can retrieve or sort the data easily.

Random access: We can get any data located at any index position.

Disadvantage of Java Array

Size Limit: We can store only fixed size of elements in the array. It doesn't grow its size at runtime. To solve this problem, collection framework is used in Java.

How to make an Array:

```
class SingleArray{

// Declaration, Instantiation, and Initialization All Together

// Static Array

    final static String myNames[] = {"Ahmad", "Sayeed", "Asmal", "Babu"};

//Non-static arrays

    String friends[] = {"Prem", "Vineet", "Javed"};

// main()

    public static void main(String...a){
```

```
        System.out.println(myNames[0]);
        // Accessing Non Satic Array
        SingleArray  sa = new SingleArray();
        System.out.println(sa.friends.length);
        System.out.println(sa.friends[1]);

        // Declaration: Declaring and Array: DataType VariableName[];
        String names[];
        int []students;
        String[] schools;
        // Instantiation. Allocating memory at heap: VariableName = new DataType[];
        schools = new String[2];
        //Initialization: VariableName[index] = Value;
        schools[0] = "Prvate School";
        schools[1] = "Goverment Schools";

        // Once declared, can't Initialize the array like this
        // students = {1,2,3,4};
    }
}
```

Create Array Of Reference variable

```
// Normal Way
        Temp t1 = new Temp(1);
        Temp t2 = new Temp(1);
        Temp t3 = new Temp(1);
// Using array and Loop
    // Declaration and Instantiation
        Temp t[] = new Temp[5];
        for(int i =0; i < t.length; i++){
```

```
            // Intialization
                t[i] = new Temp();
        }
```

Arrays Of Arrays:

Way 1

```
// Declaration
    String students[][];
// Instantiation
    students = new String[3][4];
// Initialization
    students[0][0] = "Ahmad";
    students[0][1] = "Sayeed";
    students[1][0] = "Asmal";
    students[1][1] = "Babu";
// And So On
```

Way 2

```
// Declaration
    int m[][];
// Instantiation of only Arrays
    m = new int[3][];
// Instantiation of Arrays of Arrays
    m[0] = new int[2];
    m[1] = new int[4];
    m[2] = new int[5];
// Initialization
    m[0][0] = "Ahmad";
    m[0][1] = "Sayeed";
    m[1][0] = "Asmal";
    m[2][0] = "Babu";
// How to access Arrays Of Arrays:
```

```
for ( int i = 0; i < m.length; i++ ) {
    for ( int j = 0; j < m[i].length; j++ ) {
        System.out.println( m[i][j] );
    }
}
```

Way 3

```
    int n[] = {3, 5, 7};
    // Declaration
    int d[][];
    // Instantiation of only arrays
    d = new int[3][];

    for(i=0; i<n.length; i++){
    // Instantiation of arrays of arrays using loop
    d[i] = new int[n[i]];
}
```

Way 4

```
// Declaration, Instantiation, and Initialization all together
    int s[][] = {  {10, 20, 30},  {10, 20, 30, 40},  {10, 20, 30, 40, 50}  };
```

Anonymous Array:

Arrays that we don't need to re-use.

```
// Normal Array
    int x[] = {10,20,30};
    PrintNumbers(x);
//Anonymous Array
    PrintNumbers(  new int[]{50,60,70}  );
// Another Advantage Of Anonymous Array: We can also hold reference of anonymous array. This way we can instantiate and initialize declared array with curly braces.
```

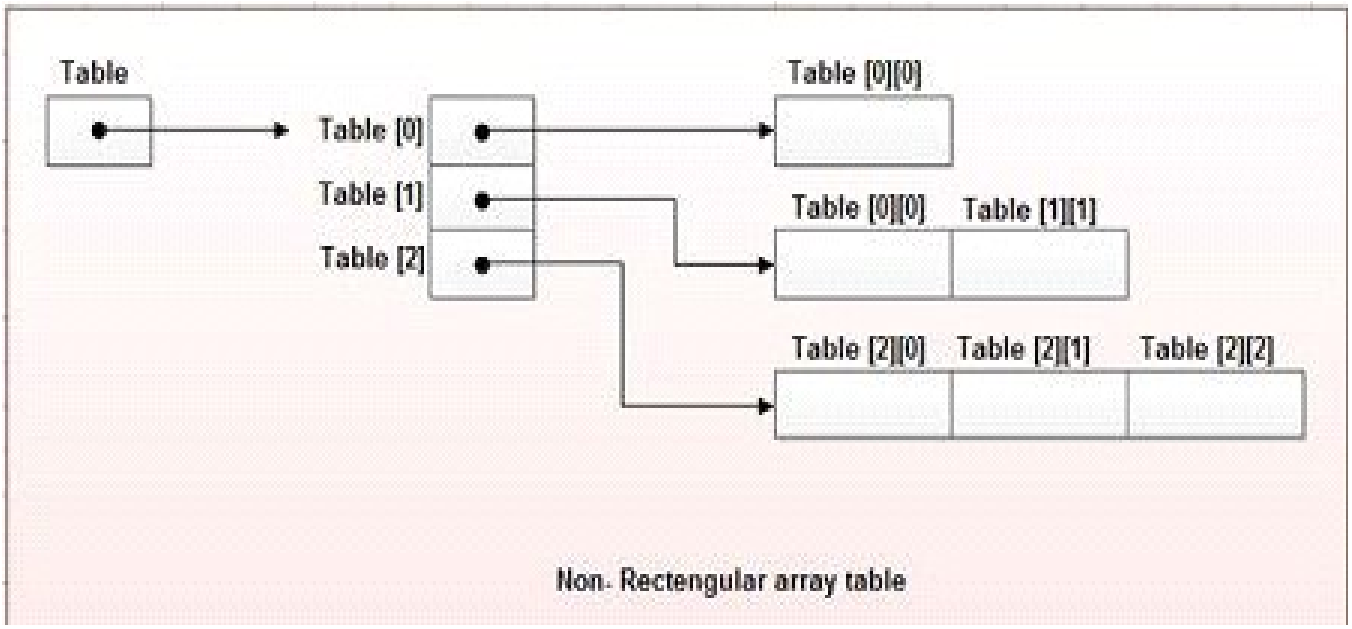
```
    int y[];
    y = new int[] {10,20,30,40,50,60};
```

Note: We can instantiate and initialize after declaration **this** way

Arrays Of Arrays - Anonymous way

```
// Normal Aray:
    int x[][] = {   {1,2,3,4},  {4,3,2,1},  {7,8,9,4}   } ;
    printArrays(x);
// Anonymous way
    printArrays( new int[][] {   {1,2,3,4},  {4,3,2,1},  {7,8,9,4}   }   );

// Create Anonymous Array Of Reference Variable
class ClassOne{
    int x = 50;
    void show(ClassOne z[]){
        for(int i =0; i< z.length; i++){
            System.out.println(z[i].x);
        }
    }
    public static void main(String...a){
        ClassOne c = new ClassOne();
    // Normal way
        ClassOne co[] = { new ClassOne(), new ClassOne(), new ClassOne() };
        System.out.println(co[1].x);
        c.show(co);
    //Anonymous way: ClassOne[] is a data type
        c.show( new ClassOne[] {
            new ClassOne(),  new ClassOne(), new ClassOne()
        } );
    }
}
```



Non- Rectengular array table