

# Java Fundamentals

Siddharth Sharma

## Arrays

- **Average of numbers using arrays:**

```
class introduction
{
    public static void main(String args[])
    {
        int array[];
        array= new int[11];
        int i;
        int x=0;
        for(i=0;i<11;i++)
        {
            array[i]=x;
            x++;
        }
        float result=0;
        int j;
        for(j=0;j<11;j++)
        {
            result= result + array[j];
        }
        System.out.println("Average is: "+result/11);
    }
}
```

- **Average of numbers using arrays:**

```
int array[]=new int[11]
```

same as

```
int array[];
array= new int[11];
```

- **Declaration and initialisation at one go:**

```
int array[]={1,2,3,4,5};
```

**Lets find average of above array:**

```
class introduction
{
    public static void main(String args[])
    {
        int array[]={1,2,3,4,5};
        int i, result=0;
        for(i=0;i<5;i++)
```

```

    {
        result= result+ array[i];
    }
    System.out.println("Average is "+result/5);
}
}

```

Here use of new was not required.

- **Introduction to multi-dimensional arrays:**

class introduction

```

{
    public static void main(String args[])
    {
        int array[][]= new int[4][5];
        int i,j;
        int x=0;
        for(i=0;i<4;i++)
        {
            for(j=0;j<5;j++)
            {
                array[i][j]=x;
                x++;
            }
        }
        for(i=0;i<4;i++)
        {
            for(j=0;j<5;j++)
            {
                System.out.print(array[i][j]+" ");
            }
            System.out.println();
        }
    }
}

```

OUTPUT:

```

0 1 2 3 4
5 6 7 8 9
10 11 12 13 14
15 16 17 18 19

```

- **When declaring multidimensional arrays, you need only specify the memory of the first (leftmost) dimension. You can allocate the remaining dimension separately.**

class introduction

```

{
    public static void main(String args[])
    {
        int array[][]= new int[5][];
        array[0]= new int[5];
        array[1]= new int[4];
        array[2]= new int[3];
    }
}

```

```

array[3]= new int[2];
array[4]= new int[1];
int i,j,x=0;
for(j=0;j<5;j++)
{
    array[0][j]=x;
    x++;
}
for(j=0;j<4;j++)
{
    array[1][j]=x;
    x++;
}
for(j=0;j<3;j++)
{
    array[2][j]=x;
    x++;
}
for(j=0;j<2;j++)
{
    array[3][j]=x;
    x++;
}
for(j=0;j<5;j++)
{
    System.out.print(array[0][j]+ " ");
}
System.out.println();
for(j=0;j<4;j++)
{
    System.out.print(array[1][j]+ " ");
}
System.out.println();
for(j=0;j<3;j++)
{
    System.out.print(array[2][j]+ " ");
}
System.out.println();
for(j=0;j<2;j++)
{
    System.out.print(array[3][j]+ " ");
}
System.out.println();
array[4][0]=x;
System.out.println(array[4][0]);
}
}

```

## OUTPUT:

```

0 1 2 3 4
5 6 7 8
9 10 11
12 13
14

```

The use of uneven (or, irregular) multidimensional arrays may not be appropriate for many applications, because it runs contrary to what people expect to find when a

multidimensional array is encountered. However, irregular arrays can be used effectively in some situations. For example, if you need a very large two-dimensional array that is sparsely populated (that is, one in which not all of the elements will be used), then an irregular array might be a perfect solution.

- **Declaration and initialisation at one go:**

```
int array[][]= { {0,1,2,3}, {4,5,6,7},{8,9,10,11},{12,13,14,15}}
```

class introduction

```
{
    public static void main(String args[])
    {
        int array[][]= { {0,1,2,3}, {4,5,6,7},{8,9,10,11},{12,13,14,15}};
        int i,j;
        for(i=0;i<4;i++)
        {
            for(j=0;j<4;j++)
            {
                System.out.print(array[i][j]+ " ");
            }
            System.out.println();
        }
    }
}
```

**OUTPUT:**

```
0 1 2 3
4 5 6 7
8 9 10 11
12 13 14 15
```

- **Alternative way to declare arrays:**

```
int array[];
```

is same as

```
int[] array;
```

```
int array1[], array2[], array3[];
```

is same as

```
int[] array1, array2, array3;
```