Arrays

Array is a collection of similar data type and which is used for store the more than one values in single variables.

How to declare the array in java

If we want to declare the array in java we have the two steps

1) Declaration of array: in declaration we specify the type of array not its size.

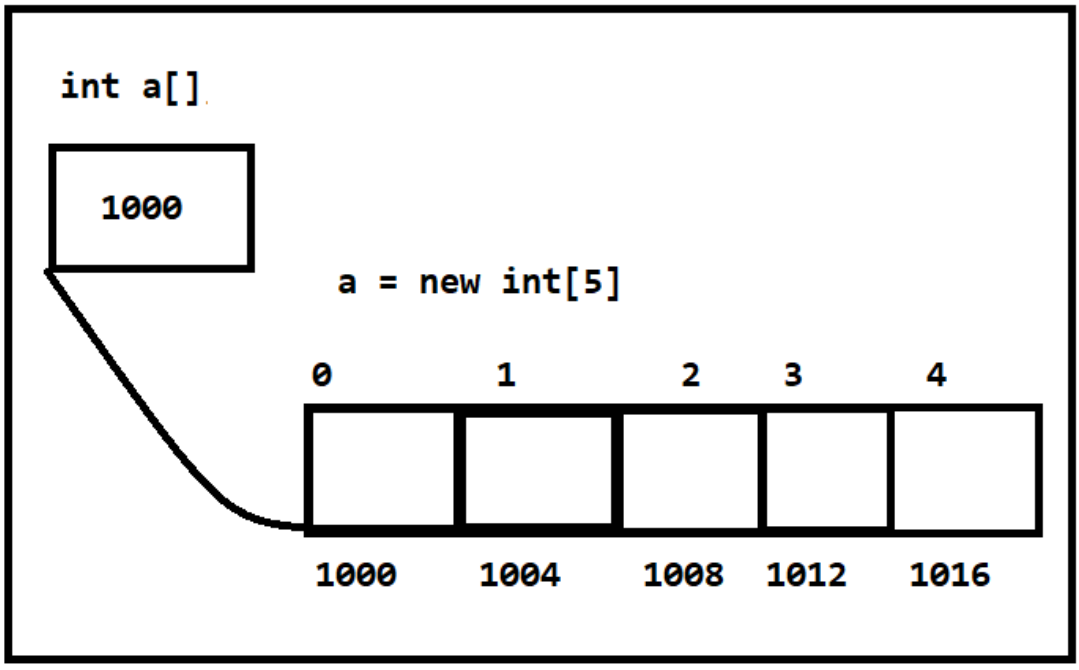
Syntax: datatype variablename []; //declaration

e.g int a[] ;//at the time declaration array variable initialize to null

2) Memory allocation of array: memory allocation means we decide the actual size of array at run time and in java array allocated by using new keyword and in java array consider as reference type or object type.

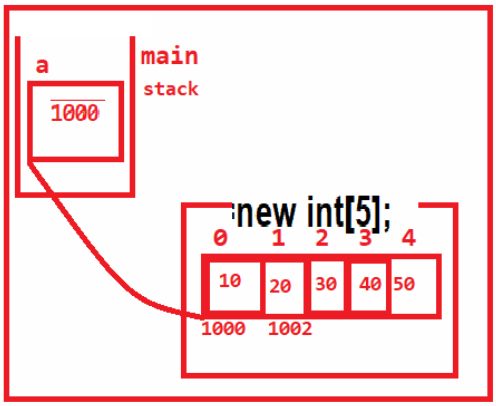
Syntax: variable name = new datatype[size];

e.g a = new int[5];



Example

```
import java.util.*;//step1
public class ArrApp
{
  public static void main(String x[])
   {
       Scanner xyz  = new Scanner(System.in);  //step2
         int a[]=new int[5];
      System.out.println("Enter the five values in array\n");
       for(int i=0; i<a.length; i++)
       {
         a[i]=xyz.nextInt(); //step3
        }
     System.out.println("display the array values");
       for(int i=0; i<a.length; i++)
       {
         System.out.printf("a[%d] --->%d\n",i,a[i]);
       }
   }
}
```



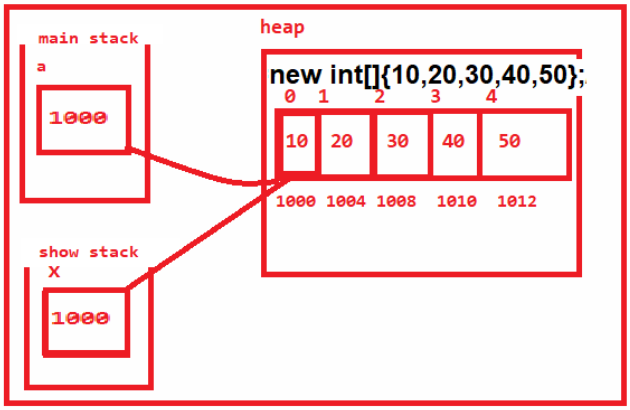
Output

```
C:\Program Files\Java\jdk1.8.0_291\bin>java ArrApp
Enter the five values in array

10
20
30
40
50
display the array values
a[0] --->10
a[1] --->20
a[2] --->30
a[3] --->40
a[4] --->50
```

Example (what will be the output)

```
import java.util.*;//step1
public class ArrApp
{
  public static void main(String x[])
  {
        int a[]=new int[]{10,20,30,40,50};//array initialization
        ArrApp   ar = new ArrApp();
        ar.show(a);
    System.out.println("Display the array\n");
       for(int i=0; i<a.length; i++)
       { System.out.printf("a[%d] -->%d\n",i,a[i]);
       }
  }
  public void show(int x[])
  {
        for(int i=0; i<x.length; i++)
         {
             x[i]=x[i]+10;
         }
   }
}
```



Example

```
Display the array

a[0] -->20
a[1] -->30
a[2] -->40
a[3] -->50
a[4] -->60
```

Description of above code

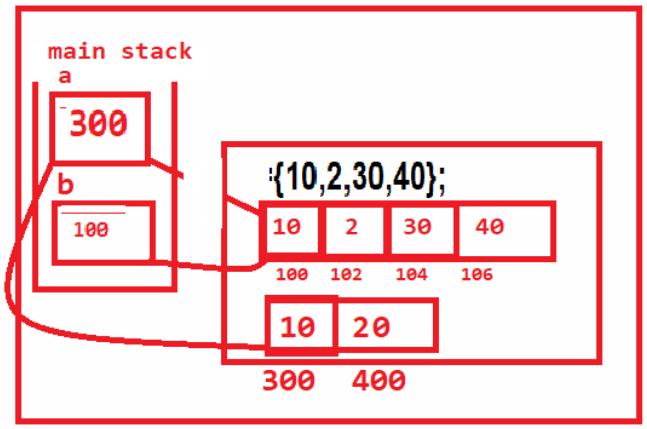
In above code we declare the array name as int a[]={10,20,30,40,50} and we pass the array to the show function when we pass array to the function then array pass its base address to the function means as per our example show function contain

x[] points to the address where a array points so if we perform any change in x[] it will reflect on array a[] so we increase the value of array by ten using x[] array in show function it will effect on the memory where array a[] points so if we perform any change in array x[] it will reflect on array a[]

Example (what will be the output of given code)

```
public class ArrApp
{
  public static void main(String x[])
   {
    int a[]={10,2,30,40};
      int b[];
     b=a;
     a=new int[]{10,20};
      for(int i=0; i<a.length;i++)
       {
         System.out.printf("%d\n",b[i]);
       }
   }

}
```



Output

```
C:\Program Files\Java\jdk1.8.0_291\bin>java ArrApp
10
2

C:\Program Files\Java\jdk1.8.0_291\bin>
```

We can declare the array in java like as datatype [] variablename;

E.g. int [] a; //valid declaration

Description of above code

If we think about the statement int a[]={10,2,30,40} means we create array in memory and its address stored in reference a[] and we have the one more statement int b[] means we create one more array name as b and we have the statement b=a means we initialize the address of a in b means b array points where array a[] points and we have statement a=new int[]{10,20} means we have the new array and reference a points to new array means a release the memory of previous created array and points to new array i.e 10 and 20 but b array points to created previously array and we have the following statement

```
for(int i=0;i<a.length;i++){
  System.out.printf("%d\n",b[i]);
```

} here we travel the loop according to size of a means 2 time because new array size is 2 and a points to the array and we print the values of b array so it will print the first two values where b array points means b array points to first array i.e 10 2 is our final output .

What is the diff between array declaration using a[] and []a ?

If we give array sub script at left hand side then all variables in that line consider as by default array e.g int [] a,b,c,d; here a b c and d all are the arrays and we give the bracket at right hand side then specific variable consider as array e.g int a[],b,c[],d; here a and c is array but b and d is normal variable.

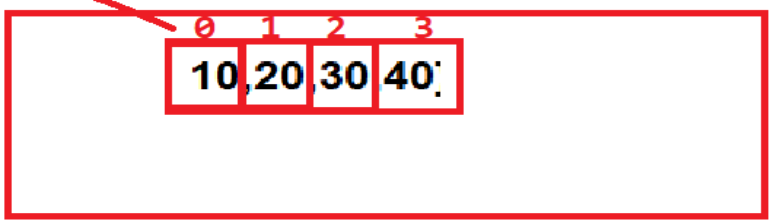
Example

```
import java.util.*;//step1
public class ArrApp
{ static int i;
  public static void main(String x[])
  {
     int []a={10,20,30,40};
  20 int b =a[++i]; a[1]
  32 int c=a[i+1]+2;
  21 int d=a[i++]+1;  20+1

  73 int e= b+c+d;
           20+32+21=73
    System.out.printf("D is %d\n",e);

  }

}
```




```
C:\Program Files\Java\jdk1.8.0_291\bin>javac ArrApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java ArrApp
D is 73

C:\Program Files\Java\jdk1.8.0_291\bin>
```

Example

Consider the following various array declarations:

int [] ar1, arr2[];

int[][] arr3;

int[] arr4[], arr5[];

Which of the following options are true?

arr2 = arr3;

arr2 = arr4;

arr1 = arr2;

arr4 = arr1;

Output

The correct answer is A and B.

Explanation: There is a difference between int[] i; and int i[]; although in both the cases i is an array of integer values. Therefore the correct option is A and B as they are the declarations of the integer. The basic difference is that if you declare multiple variables in the same statement such as int [] i, j; and int i[], j; then it Implies that i and j are not of the same type.

Example

Imagine you need to handle the records of multiple students and declaring a separate variable and then assigning the values will become a tedious task. Therefore, you write the following program to implement the concept of arrays which has simplified your task:

Source code

```
public class Ques
 { public static void main(String args[]) {
  String[][][] arr = {
                        {
                         { "Suchita", "Vikash" , "Deepak"},
                         { "Charu", null, "Shikha"}
                        },
                       {
                          {"Shalini"}, {null}
                        },
                       {
```

```
                    {"Hemal"}
                },
                {
                    { "Santosh", "Manish"}, {}
                }
            };
        System.out.println(arr[0][1][1]);
 }
}
```

What will be the output after compilation and execution of the preceding program?

A. The program will throw the runtime exception.

B. The program will throw ArrayIndexOutOfBoundsException.

C. The program will display null.

D. The program will compile successfully but it will not display anything.

Output

The correct option is C.

Explanation: The program deals with three dimensional array and will display null as following is the structure of elements assigned to the arr array.

arr[0][0][0] = Suchita

arr[0][0][1] = Vikash

arr[0][0][2] = Deepak

arr[0][1][0] = Charu

arr[0][1][1] = null

arr[0][1][2] = Shikha

arr[1][0][0] = Shalini

arr[1][1][0] = null

arr[2][0][0] = Hemal

arr[3][0][0] = Santosh
arr[3][0][1] = Manish
Therefore, if you try to print the value of the arr[0][1][1] element of the arr array, then the null value will be displayed

Example

Sam works in Xyz Company as Java programmer and he designed the following program:

Source Code

```
class Rose{
public void sam() {
 int y[] = {4, 2, 8};
 for (int x=2; x<1+3*2-4; x++){
 System.out.print(x+" ");
 for (int j:y) {
     j=j*x-4;
 System.out.print(j+" ");
 }
 }
 }
 public static void main(String[] args) {
 Rose r = new Rose();
 r.sam();
 }
}
```

What would be the output of this program? Choose the correct option from the following:

A. The program displays 2 4 2 8 B. The program displays 2 4 0 12
C. The program displays 2 4 4 16 D. The program displays 3 4 0 12

Correct option is B.

Explanation: Correct option is B. Firstly the value of the variable x is 2, which is less than the 3 (Value specified in expression) therefore control will transfer into for each loop and prints the array variable with Modifications according to the expression. Array element is fetched into j. First array element is 4 and the Expression is j*x-4 i.e. 4*2-4=4. In the same way, other array elements are extracted and displayed with modifications.

Example

Sam as a developer in Dkinfotech created the following program

```
class Rose {
 static int j;
 public int arr() {
 int y[] = { 5 , 7, 8 , 6};
 j = y[2]; return j;
 }
 public static void main (String args[]) {
 Rose r=new Rose();
 int x = r.arr( );
 System.out.println(x);
 switch(x) {
 case 0: System.out.print(0 + " ");break;
 case 2: System.out.print(2 + " ");break;
 case 8: System.out.print(8 + " ");
```

```
 case 5: System.out.print(5 + " ");break;
 default: System.out.print("Default");
 }
 }
}
```

What would be the output when Sam compile and execute this program?

A. Program will not compile successfully B. Program will display 8 8 5

C. Program will display 8 5 D. Program will display 8

Option B is the correct answer.

Explanation: Option B is the correct answer. Firstly the control will transfer to arr () method and retrieves the array element at array [2] index and returns that value to calling routine. In the calling routine the retrieved value is first displayed and then used in switch statement. Therefore, the result is 8 8 and 5 is displayed because the case 8 does not have break and 5 is the statement written in next case. Therefore, options C and D are incorrect. Option A is incorrect because program will successfully compile.

Two Dimensional Arrays

Two dimensional array is used for create the matrix but it is not a matrix internally.

Syntax:

```
datatype  variablename[][];
or
datatype [][]variablename;
or
datatype []variablename[];
e.g int []a,b[][],c[];
```

//here a is single dimension b is three dimension and c is two dimension

Memory allocation

variablename= new datatype[size][size];

Example

```
import java.util.*;
public class MatrixApplication
{
   public static void main(String x[])
   {
      Scanner xyz  = new Scanner(System.in);
       int a[][]=new int[3][3];
     System.out.println("Enter the values in matrix");
      for(int i=0; i<a.length; i++)
     {
            for(int j=0; j<a[i].length; j++)
            {    a[i][j]=xyz.nextInt();
            }
     }
     System.out.println("display the matrix");
      for(int i=0; i<a.length; i++)
      {
             for(int j=0; j<a[i].length; j++)
             { System.out.printf("%d\t",a[i][j]);
            }
       System.out.printf("\n");
      }
   }
```

```
}
```

Jagged Array

Jagged Array is facility in java where we can create the matrix every row having different column list called as jagged array.

Example we want to create the matrix like as

```
1   2    3
4   5     6   7
8   9
```

Syntax :

```
datatype variablename[][]=new datatype[rowsize][];
 variablename[rowindex]=new datatype[colsize];
```

Example

```
import java.util.*;
public class MatrixApplication{
   public static void main(String x[])
   {
       Scanner xyz  = new Scanner(System.in);
        int a[][]=new int[3][];
          a[0]=new int[3];
          a[1]=new int[4];
          a[2]=new int[2];
      System.out.println("Enter the values in matrix");
       for(int i=0; i<a.length; i++)
      {
            for(int j=0; j<a[i].length; j++)
            {    a[i][j]=xyz.nextInt();
```

```
            }
      }
      System.out.println("display the matrix");
       for(int i=0; i<a.length; i++)
       {    for(int j=0; j<a[i].length; j++)
              { System.out.printf("%d\t",a[i][j]);
              }
        System.out.printf("\n");
       }
   }
}
```