**Arrays:**

Array is collection of similar data types

Starting point of index is from 0(zero), 1….

Advantages – Code optimization (retrieve or sort the data efficiently), random access (get data located at an index position)

Disadvantages – size limit (it doesn’t grow size at runtime to solve this we use collections)

Two types – Single dimensional, Multidimensional array

Single dimensional array:

Syntax to declare

Datatype[] arr; or

Datatype []arr; or

Datatpe arr[]

Example:

**package** com.lokesh;

**public** **class** HelloWorld2{

**public** **static** **void** main(String args[]) {

// **TODO** Auto-generated method stub

**int** a[] = **new** **int**[3]; //declaration and instantiation

a[0] = 10; //initialization

a[1] = 20;

a[2] = 30;

**for**(**int** i =0;i<a.length;i++) { //length is property of array

System.***out***.println(a[i]);

}

}

}

Another example:

**package** com.lokesh;

**public** **class** HelloWorld2{

**public** **static** **void** main(String args[]) {

// **TODO** Auto-generated method stub

**int** a[] = {10,20,30}; //declaration, instantiation and initialization

**for**(**int** i =0;i<a.length;i++) { //length is property of array

System.***out***.println(a[i]);

}

}

}

Example using for-each loop:

**package** com.lokesh;

**public** **class** HelloWorld2{

**public** **static** **void** main(String args[]) {

// **TODO** Auto-generated method stub

**int** a[] = {10,20,30}; //declaration, instantiation and initialization

**for**(**int** i : a) { //length is property of array

System.***out***.println(i);

}

}

}

Passing Array to a method: We can pass the java array as parameter to a method so that we can re-use the same logic on any array

Below is an example to get minimum number of an array using method

**package** com.lokesh;

**public** **class** HelloWorld {

**public** **void** min(**int** arr[]) {

**int** min =arr[0];

**for**(**int** i=1;i<arr.length;i++) {

**if**(min>arr[i]) {

min =arr[i];

}

}

System.***out***.println(min);

}

}

**package** com.lokesh;

**public** **class** HelloWorld2{

**public** **static** **void** main(String args[]) {

// **TODO** Auto-generated method stub

**int** a[] = {6,45,27,2,23};

HelloWorld hw =**new** HelloWorld();

hw.min(a);

}

}

Anonymous array: you don’t need to declare array while passing an array to method

Example:

**package** com.lokesh;

**public** **class** HelloWorld {

**public** **void** min(**int** arr[]) {

**for**(**int** i=0;i<arr.length;i++) {

System.***out***.println(arr[i]);

}

}

}

**package** com.lokesh;

**public** **class** HelloWorld2{

**public** **static** **void** main(String args[]) {

// **TODO** Auto-generated method stub

HelloWorld hw =**new** HelloWorld();

hw.min(**new** **int**[] {2,4,7,1,6}); //passing anonymous array to method

}

}

Returning array from method: Below is an example for returning the data as array where the method add5 just add 5 for the parameters it received and sends back the array

**package** com.lokesh;

**public** **class** HelloWorld {

**public** **int**[] add5(**int** arr[]) {

**for**(**int** i=0;i<arr.length;i++) {

**int** min =arr[i];

min = min + 5;

arr[i] =min;

}

**return** arr;

}

}

**package** com.lokesh;

**public** **class** HelloWorld2{

**public** **static** **void** main(String args[]) {

// **TODO** Auto-generated method stub

HelloWorld hw =**new** HelloWorld();

**int** a[] = {3 ,5,8};

**int** b[] = hw.add5(a);

**for**(**int** j=0;j<b.length;j++) {

System.***out***.println(b[j]);

}

}

}

ArrayIndexoutofbound exception: if we try to retrieve or show the array value for length of array size is equal or greater than array size we get this exception

Example:

**package** com.lokesh;

**public** **class** HelloWorld2{

**public** **static** **void** main(String args[]) {

// **TODO** Auto-generated method stub

**int** a[] = {3,5,8,13};

**for**(**int** j=0;j<=a.length;j++) {

System.***out***.println(a[j]);

}

}

}

Multidimensional array: data stored in row and column based index (matrix form)

Syntax to declare:

Datatype[][] arr; or

Datatype [][]arr; or

Datatype arr[][]; or

Datatype []arr[];

Instantiation:

Int arr[][] =new int[3][3];

Initialization:

arr[0][0] = 1;

arr[0][1] = 2;

arr[0][2] = 3;

arr[1][1] = 4;

and so on;

Below declaration, instantiation and initialization

int arr[][] = {{1,2,3},{4,5,6},{7,8,9}}

Example of multidimensional array:

**package** com.lokesh;

**public** **class** HelloWorld2{

**public** **static** **void** main(String args[]) {

// **TODO** Auto-generated method stub

**int** a[][] = {{1,2,3},{4,5,6},{7,8,9}};

**for**(**int** i=0;i<a.length;i++) {

**for**(**int** j=0;j<a.length;j++) {

System.***out***.println(a[i][j]);

}

}

}

}

Jagged array: means columns are different like below

0 1 2

3 4 5 6

7 8

Below is an example for copying an array

**package** com.lokesh;

**public** **class** HelloWorld2{

**public** **static** **void** main(String args[]) {

// **TODO** Auto-generated method stub

**int** a[] = {1,2,3,4};

**int** b[] = **new** **int**[4];

System.*arraycopy*(a, 0, b, 0, 4);

**for**(**int** i=0;i<b.length;i++) {

System.***out***.println(b[i]);

}

}

}

Below is an example for cloning an array

**package** com.lokesh;

**public** **class** HelloWorld2{

**public** **static** **void** main(String args[]) {

// **TODO** Auto-generated method stub

**int** a[] = {1,2,3,4};

**int** b[] = a.clone();

**for**(**int** i=0;i<b.length;i++) {

System.***out***.println(b[i]);

}

}

}