**Array**

* It is a group of variables of similar data type i.e. homogenous data structure referred by single name.
* It elements are stored in a continuous memory location.
* The size of array should be mentioned while declaring it.
* Array elements are always counted from zero (0) onward.
* Array elements can be accessed using the position of the element in the array.



**Operation on array**

* Add new element to an array.
* Delete element from an array.
* Search element from an array.
* Sort array element from an array.
* Compare Two Arrays.  etc

**Advantages**

* Code Optimization: It makes the code optimized; we can retrieve or sort the data efficiently.
* Random access: We can get any data located at an index position.

**Disadvantages**

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

**Types of Array in java**

There are two types of array.

* Single Dimensional Array
* Multidimensional Array ( 2D and 3D Array)

**Single Dimensional Array in Java**

**Syntax:**

|  |
| --- |
| **dataType[] variable\_name; (or)**  **dataType []variable\_name; (or)**  **dataType variable\_name[];** |

The dataType specify the basic data type of array elements, which specify what type of elements can be stored in array.

Eg: the following array can be used to store “integer” elements.

|  |
| --- |
| int no[]; |

Declaration just specifies that array “no” can be used to store integers but no memory allocation take place at the time of declaration.

**Memory allocation or** **Instantiation of an Array in Java**

Memory allocation is done using operator “new” in java.

|  |
| --- |
| Syntax: |
| var = new dataType[size]; |

The variable “var” is the name of the array. The dataType referred to the dataType of the elements that can be stored in the array. “size” refers to the number of elements for which the memory is to be allocated dynamically.

|  |
| --- |
| Eg: following array can store 20 elements. |
| no=new int[20]; |

The above steps can be combined into one step as follows:

|  |
| --- |
| int no[]=new int[20]; |

**Q: Develop a java program to accept “N” numbers of array elements from the user and print the sum of all array elements.**

Ans:

class AddElement

{

public static void main(String args[])

{

int i,no,a[],sum=0;

try{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the size of array:");

no=Integer.parseInt(br.readLine());

a=new int[no];

System.err.println("Enter the array elements one by one:");

for(i=0;i<no;i++)

{

a[i]=Integer.parseInt(br.readLine());

}

System.out.println("");

for(i=0;i<no;i++)

{

sum=sum+a[i];

}

System.out.println("Sum="+sum);

}catch(Exception ex){ex.printStackTrace();}

}

}

**Q: Write a java program to accept “N” number of array elements from the user and print its ODD number and EVEN number separately.**

Class Odd\_Even

{

public static void main(String args[])

{

int i,no,a[],sum=0;

try{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the size of array:");

no=Integer.parseInt(br.readLine());

a=new int[no];

System.err.println("Enter the array elements one by one:");

for(i=0;i<no;i++)

{

a[i]=Integer.parseInt(br.readLine());

}

System.out.print("EVEN NO:");

for(i=0;i<no;i++)

{

if(a[i]%2==0)

System.out.print("\t"+a[i]);

}

System.out.println("");

System.out.print("ODD NO:");

for(i=0;i<no;i++)

{

if(a[i]%2!=0)

System.out.print("\t"+a[i]);

}

}catch(Exception ex){ex.printStackTrace();}

}

Q: **How do you find the duplicate number on a given integer array?**

|  |
| --- |
| **public class DuplicateElement {**  **public static void main(String[] args) {**    **//Initialize array**  **int [] arr = new int[] {1, 2, 3, 4, 2, 7, 8, 8, 3};**    **System.out.println("Duplicate elements in given array: ");**  **//Searches for duplicate element**  **for(int i = 0; i < arr.length; i++) {**  **for(int j = i + 1; j < arr.length; j++) {**  **if(arr[i] == arr[j])**  **System.out.println(arr[i]);**  **}**  **}**  **}**  **}** |

Q: Write a java program to accept “N” number of array elements from the user.

Find the maximum and minimum number from the array elements.

Q: Write a java program to accept “N” number of array elements from the user and perform following operation.

* Insert new element to the array.
* Delete element from an array.

|  |
| --- |
| import java.io.\*;  public class AddElement {    public void add()  {  int no,i,newno,pos;  int a[]=new int[10];  try  {  BufferedReader br=new BufferedReader(new InputStreamReader(System.in));  System.out.println("Enter the size of array:");  no=Integer.parseInt(br.readLine());  System.out.println("Enter the elements one by one:");  for(i=0;i<no;i++)  {  a[i]=Integer.parseInt(br.readLine());  }    System.out.println("Enter the New Number:");  newno=Integer.parseInt(br.readLine());    System.out.println("Enter the position:");  pos=Integer.parseInt(br.readLine());    for(i=no-1;i>=pos;i--)  {  a[i+1]=a[i];  }  a[i+1]=a[i];  a[pos-1]=newno;  no++;  for(i=0;i<no;i++)  {  System.out.println(""+a[i]);  }  }catch(Exception ex){ex.printStackTrace();}  }    public static void main(String[] args) throws IOException{  AddElement a=new AddElement();  a.add();  }    } |

|  |
| --- |
| import java.io.\*;  public class DeleteElement {      public static void main(String[] args) {    int no,i,pos;  int a[]=new int[10];  try  {  BufferedReader br=new BufferedReader(new InputStreamReader(System.in));  System.out.println("Enter the size of array:");  no=Integer.parseInt(br.readLine());  System.out.println("Enter the elements one by one:");  for(i=0;i<no;i++)  {  a[i]=Integer.parseInt(br.readLine());  }  System.out.println("Enter the posistion to delete the element:");  pos=Integer.parseInt(br.readLine());    for(i=pos-1;i<no;i++)  {  a[i]=a[i+1];  }  no--;  for(i=0;i<no;i++)  {  System.out.println(""+a[i]);  }  }catch(Exception ex){ex.printStackTrace();}  }  } |

Q: Write a java program to accept “**N** “number of integer array from the user and search the number in array using **Linear search**.

Q: Write a java program to accept “**N** “number of integer array from the user and search the number in array using **Binary search**.

Q: Write a java program to accept “**N** “number of integer array from the user and print the number in sorted order using selection sort.

Ans:

class DemoSelectionSort

{

public static void main(String args[])

{

int i,j,temp=0,a[],no;

try{

br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the Size of array:");

no=Integer.parseInt(br.readLine());

a=new int[no];

System.out.println("Enter the array elements one by one:");

for(i=0;i<no;i++)

a[i]=Integer.parseInt(br.readLine());

for(i=0;i<no-1;i++)

{

for(j=i+1;j<no;j++)

{

if(a[i]>a[j])

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

System.out.println("");

for(i=0;i<no;i++)

{

System.out.print("\t"+a[i]);

}

System.out.println("");

}catch(Exception ex){ex.printStackTrace();}

}

}

Q: Write a java program to accept “**N** “number of integer array from the user and print the number in sorted order using array using bubble sort.

Ans:

class DemoBubbleSort

{

public static void main(String args[])

{

int i,j,temp=0,a[],no;

try{

br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the Size of array:");

no=Integer.parseInt(br.readLine());

a=new int[no];

System.out.println("Enter the array elements one by one:");

for(i=0;i<no;i++)

a[i]=Integer.parseInt(br.readLine());

for(i=0;i<no-1;i++)

{

for(j=0;j<no-1;j++)

{

if(a[j]>a[j+1])

{

temp=a[j];

a[j]=a[j+1];

a[j+1]=temp;

}

}

} System.out.println("");

for(i=0;i<no;i++)

{

System.out.print("\t"+a[i]);

}

System.out.println("");

}catch(Exception ex){ex.printStackTrace();}

}

}

**Passing Array to a Method in Java**

We can pass the java array to method so that we can reuse the same logic on any array.

Let's see the simple example to get the maximum number of an array using a method.

import java.io.\*;

class Maximum

{

void find\_max(int no,int x[])

{

int i,max=0;

for(i=0;i<no;i++)

{

if(x[i]>max)

max=x[i];

}

System.out.println("Max:\t"+max);

}

public static void main(String args[])

{

int i,no,a[],max;

try{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the size of array:");

no=Integer.parseInt(br.readLine());

a=new int[no];

System.err.println("Enter the array elements one by one:");

for(i=0;i<no;i++)

{

a[i]=Integer.parseInt(br.readLine());

}

Sample s=new Sample();

s.find\_max(no, a);

}catch(Exception ex){ex.printStackTrace();}

}

}

}

**Returning Array from the Method**

import java.io.\*;

class DemoReturnArray{

int no,i,a[];

//creating method which returns an array

int[] get()

{

try{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the size of array:");

no=Integer.parseInt(br.readLine());

a=new int[no];

System.err.println("Enter the array elements one by one:");

for(i=0;i<no;i++)

{

a[i]=Integer.parseInt(br.readLine());

}

}catch(Exception ex){ex.printStackTrace();}

return a;

}

public static void main(String args[])

{

DemoReturnArray s=new DemoReturnArray();

//calling method which returns an array

int no[]=s.get();

int i;

try{

//printing the values of an array

for(i=0;i<no.length;i++)

{

System.out.println(""+no[i]);

}

}catch(Exception ex){ex.printStackTrace();}

}

}

**Multidimensional Array in Java**

You can create an array of an array is known as multidimensional array. In multidimensional array, data is stored in row and column based index (also known as matrix form).

|  |
| --- |
| **A two dimensional array is declared as:** |
| dataType variable\_name[][]; (or)  dataType [][]variable\_name; (or)  dataType[][] variable\_name; |

For example the following array can be used to store integer elements.

|  |
| --- |
| int a[][]; |

Declaration just specifies that array “a” can be used to store integers but no memory allocation take place.

**Memory Allocation**

The general form of the operator “new” is as follows:

|  |
| --- |
| var = new dataType[rows][columns]; |

The “var” is the name of two-dimensional array. The dataType refers to the dataType of the elements that can be stored in the array. The first index indicates the number of rows and the second index indicates the number of columns in the two-dimensional array to be allocated dynamically.

|  |
| --- |
| Eg: following array can store a table consisting of 5 rows and 4 columns. |
| A=new int[5][4] |

.  
The above mentioned steps can be combined into one as follows:

|  |
| --- |
| int a[][]=new int[5][4]; |

**//Java Program to illustrate the use of multidimensional array**

class Demo{

public static void main(String args[]){

//declaring and initializing 2D array

int a[][]={{10,20,30},{20,40,50},{45,48,52}};

//printing 2D array

for(int i=0;i<3;i++){

for(int j=0;j<3;j++){

System.out.print(a[i][j]+" ");

}

System.out.println();

}

}}

**Q: Write a java program to accept 3X3 matrix from the user and print the same.**

**import java.io.\*;**

**public class Demo\_2D {**

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

**{**

**try**

**{**

**int a[][]=new int[3][3];**

**int i,j;**

**BufferedReader br=new BufferedReader(new InputStreamReader(System.in));**

**System.out.println("Enter 3X3 Matrix:");**

**for(i=0;i<3;i++)**

**{**

**for(j=0;j<3;j++)**

**{**

**a[i][j]=Integer.parseInt(br.readLine());**

**}**

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

**}**

**for(i=0;i<3;i++)**

**{**

**for(j=0;j<3;j++)**

**{**

**System.out.print("\t"+a[i][j]);**

**}**

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

**}**

**}catch(Exception ex){ex.printStackTrace();}**

**}**

**}**

**Q: Write a java program to accept 3X3 matrix from the user and** print**the**transpose**of a given**matrix**.**

**import java.io.\*;**

**public class TransposeMatrix {**

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

**{**

**try**

**{**

**int a[][]=new int[3][3];**

**int i,j;**

**BufferedReader br=new BufferedReader(new InputStreamReader(System.in));**

**System.out.println("Enter the Rows:");**

**r=Integer.parseInt(br.readLine());**

**System.out.println("Enter the Columns:");**

**c=Integer.parseInt(br.readLine());**

**for(i=0;i<r;i++)**

**{**

**for(j=0;j<c;j++)**

**{**

**a[i][j]=Integer.parseInt(br.readLine());**

**}**

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

**}**

**for(i=0;i<r;i++)**

**{**

**for(j=0;j<c;j++)**

**{**

**System.out.print("\t"+a[j][i]);**

**}**

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

**}**

**}catch(Exception ex){ex.printStackTrace();}**

**}**

**}**

**Q: Write a java program to accept one matrix from the user and** print**the row wise addition of the matrix.**

**Q: Write a java program to accept one matrix from the user and** print**the column wise addition of the matrix.**

**Q: Write a java program to accept one matrix from the user and** check **the** matrix is diagonal matrix or not**.**

**Q: Write a java program to accept one matrix from the user and** check **the** matrix is symmetric matrix or not**.**

**Q: Write a java program to accept one matrix from the user and** check **the** matrix is skew matrix or not**.**

Write a Java Program to find the sum of each row and each column of a matrix