**DAY-05**

**09/05/25**

**ARRAYS**

**Array**: collection of similar type of data in a continuous memory allocation is called Array.

* In C and C++ we have pointers so definitely elements will be stored in continuous format
* In java we don’t have pointers then there is a change of elements might store in random places

Syntax:

datatype[] array name

Premitive Data Type & Non Premiive Data Type:

**Primitive data types cant be splitted or divided**

**Non-Premitive data types can be divided.**

**Example for Pre**: int float char

**Example for non-Primitive data type** :String,Array,all the object that created by users

class object{

int obj;

node next;

}

All the objects will be stored in heap memory

All the reference will be stack memory

Stack memory having less space comparing to heap

All the arrays are stores in same datatype instead of creating multiple variable create an array

Int a=10

Int b=20

Int c=30

Int d=40

int[] arr={10,20,30,40}

If we create a multiple variables to get the data simply we can print the variable when it comes to array if we want to print the data we can print using index values and by default index values start from 0.

public static void main(String[] args) {

int[] arr={1,2,3,4,5};//Deceleration and Initialization

}

public class Main{

public class void main(String args[]){

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

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

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

}

}

}

To get the length of the array we have a method called array.length

**In arrays it is array\_name.length and in string it is string\_name.length()**

**Camel Case and Snake Cake:**

arrayName ##camel case

String\_name ##Snake case

In java it is preferred to use camel case to write production level code

public class Main{

public static void main(String args[]){

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

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

System.out.println(arr[i] + " ");

}

}

}

**Types of Printing format:(FOR EACH LOOP)**

public class Main{

public static void main(String args[]){

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

for(int i:arr){

System.out.print(i+" ");

}

}

}

#OUTPUT:1 2 3 4 5

In for each loop we don’t need index values

**To string is an inbuilt method to print an array.**

import java.util.\*;

class Main{

public static void main(String args[]){

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

System.out.print(Arrays.toString(arr));

}

}

Output:[1,2,3,4,5]

Two sting can accept only one argument

**Deceleration the size of an array initializing the values later:**

**Array having fixed size if we declare the array size it cant be change**

import java.util.\*;

class Main{

public static void main(String args[]){

int[] arr = new int[5];

System.out.println((arr[0]));

}

}

Before=array reference was created with the help of new keyword actual object will be created in the heap memory for that particular data type and size.

If the array type is int all the default values are zeros if it a string all the default values are null

And null is a literal.

**Taking input form user and making array print:**

import java.util.\*;

class Main{

public static void main(String args[]){

Scanner in=new Scanner(System.in);

int[] arr = new int[5];

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

System.out.println("Enter the values:-");

arr[i]=in.nextInt();

}

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

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

}

for(int i:arr){

System.out.print(i);

}

System.out.print(Arrays.toString(arr));

}

}

Output:

Enter the values:-

1

Enter the values:-

2

Enter the values:-

3

Enter the values:-

4

Enter the values:-

5

1

2

3

4

5

12345[1, 2, 3, 4, 5]

**Printing Even Odd numbers taking input from user:**

import java.util.\*;

class Main {

public static void main(String args[]) {

Scanner in = new Scanner(System.in);

int[] arr = new int[5];

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

System.out.println("Enter the values:-");

arr[i] = in.nextInt();

}

System.out.print("Even numbers: ");

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

if (arr[i] % 2 == 0) {

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

}

}

System.out.print("\nOdd numbers: ");

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

if (arr[i] % 2 != 0) {

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

}

}

}

}

Output:

Enter the values:-

1

Enter the values:-

2

Enter the values:-

3

Enter the values:-

4

Enter the values:-

5

Even numbers: 2 4

Odd numbers: 1 3 5

**To Print Highest value from an array:**

import java.util.\*;

class Main {

public static void main(String args[]) {

Scanner in = new Scanner(System.in);

int[] arr = new int[5];

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

System.out.println("Enter the values:-");

arr[i] = in.nextInt();

}

int highest = Integer.MIN\_VALUE;

int secondHighest = Integer.MIN\_VALUE;

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

if (arr[i] > highest) {

secondHighest = highest;

highest = arr[i];

} else if (arr[i] > secondHighest && arr[i] != highest) {

secondHighest = arr[i];

}

}

if (secondHighest == Integer.MIN\_VALUE) {

System.out.println("There is no second highest value.");

} else {

System.out.println("The second highest value is: " + secondHighest);

}

}

}

Output:

Enter the values:-

1

Enter the values:-

2

Enter the values:-

3

Enter the values:-

4

Enter the values:-

5

The second highest value is: 4

Print Highest Value:

import java.util.\*;

class Main {

public static void main(String args[]) {

Scanner in = new Scanner(System.in);

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

int heightValue=arr[0];

int secondHeightValue=arr[0];

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

if(arr[i]>heightValue){

heightValue=arr[i];

}

}

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

if(arr[i]>secondHeightValue){

heightValue=arr[i];

}

}

System.out.println(heightValue);

}

}

Output:5

**Second Highest Number to print:**

import java.util.\*;

class Main {

public static void main(String args[]) {

Scanner in = new Scanner(System.in);

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

int heightValue=arr[0];

int secondHeightValue=arr[0];

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

if(arr[i]>heightValue){

heightValue=arr[i];

}

}

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

if (arr[i] != heightValue && arr[i] > secondHeightValue) {

secondHeightValue = arr[i];

}

}

System.out.println(secondHeightValue);

}

}

**Output:4**

Check whether given given array is sorted or not

Check whether the element is there are not

[1,0,1,0,1,0]Move all the zeros to right side.[1,1,1,0,0,0]