Array

Non-primitive datatype

To store multiple values of similar datatype arrays are used

Int n1,n2,n3,n4,n5;

Int n1,n2,n3,n4,n5,n6…………………

Declaration => Datatype arrayname; => int ar[];

Declaration & Memory allocation => int ar[]=new int[5];

Assigning the values ar[0]=2;ar[1]=3000;ar[2]=-400;ar[3]=0;ar[4]=20000;

Arrayname => ar

Size => 5

Datatype => int

Index => 0 1 2 3 4

Elements => ar[0], ar[1], ar[2], ar[3], ar[4]

Values => 2, 3000, -400, 0, 20000

ar[0] ar[1] ar[2] ar[3] ar[4] elements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 3000 | -400 | 0 | 20000 |

0 1 2 3 4 index

Declaration, memory allocation and assignment together

Int ar[]={2,3000,-400,0,20000};

In case of array the values are stored in continuous memory locations, thus allowing access to all the values using common name (arrayname) and position (index). Each value can be managed individually using array element arrayname[index]

How are you?

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| H | o | w |  | a | r | e |  | y | o | u | ? |

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

12

Array of objects

Emp e[]; // refvar array declaration for type Emp (class or UDD)

e=new Emp[2]; // memory allocation for refvar array and size specification

arrayname => e

datatype => Emp

size => 2

index => 0 1

element => e[0] e[1]

|  |  |
| --- | --- |
|  |  |

e[0]=new Emp(1,"Suman",67000.0f); // creating object of Emp type and storing ref in refvar e[0]

Eno 1

Ename Suman

Sal 67000.0

Double dimensional array is associated with the concept of table/ rows& columns

Closely associated with nested for loops

Outer loop is generally used for rows and inner loop for columns

Int da[3][4]

Arrayname =>da

Datatype => int

Size => no. of rows=3, no. of column=4

Index => 0 0 , 0 1, 0 2, 0 3

1 0, 1 1, 1 2, 1 3

2 0, 2 1, 2 2, 2 3

Rows

|  |  |  |  |
| --- | --- | --- | --- |
| Da[0][0] 80 | Da[0][1] 9 | Da[0][2] -5 | Da[0][3] 600 |
| Da[1][0] | Da[1][1] | Da[1][2] | Da[1][3] |
| Da[2][0] | Da[2][1] | Da[2][2] | Da[2][3] |

Column