OBJECT ORIENTED PROGRAMMING USING JAVA

OUTLINE

- Array
- Why to use Array?
- Array: Advantages and Limitation
- Declaring an Array
- Array construction
- Types of Array
- Single Dimensional Array
- Multi Dimensional Array

ARRAY

- Array is a collection of elements of similar data types.
- The size (length) of an array is fixed. It cannot be changed after defining.
- The size of an array must be specified by an int value.
- The first element of an array starts with index zero.
- An element of an array is accessed by its respective index.
- Java array can be also used as a static field, a local variable or a method parameter .

ARRAY(CONT..)

104	532	703	451	673	983
index[0]	index[1]	index[2]	index[3]	index[4]	index [5]

- Array length: 6
- First index: 0
- Last index: 5

WHY TO USE ARRAY?

- Suppose I want one integer value

Sol: int x=1;

- Suppose I want two integer value

Sol: int x1=1, x2=2;

- Suppose I want 1000 integer value

Sol: int x1=1, x2=2,……..x1000=1000;

Hence readability of code becomes down[because for 1000 values 1000 variables are required]

Solution of this problem is an array

int [] x = new int [1000];

By using index we can access the all values of array

ARRAY: ADVANTAGES AND LIMITATION

- **DEFINATION**: **An array is a indexed element of fixed number of homogenous data element.**
- **ADVANTAGE**
- The main advantage of array is that we can represent multiple values under the same name.
- **Limitation**
- Once we have created an array there is no chance of increasing and decreasing size based on our requirement, hence memory point of view array is not recommended.
- Array can hold only homogeneous data type.
- But we can resolve these issues using collections.

DECLARING AN ARRAY

- **Single dimensional array:**
- int[] a; (recommended)
- int a[];
- int []a;
- **Two dimensional array:**
- int[][] a;
- int [][]a;
- int   a[][];
- **Three dimensional array:**
- int[][][] a;
- int [][][]a;
- int   a[][][];

Which of the following are valid declaration?

1. int[] a,b;
2. int[] a[],b;
3. int[] []a,b;
4. int[] []a,[]b;

DECLARING AN ARRAY

- **Single dimensional array:**
- int[] a; (recommended)
- int a[];
- int []a;
- **Two dimensional array:**
- int[][] a;
- int [][]a;
- int a[][];
- **Three dimensional array:**
- int[][][] a;
- int [][][]a;
- int a[][][];

Which of the following are valid declaration?

1. int[] a,b;
2. int[] a[],b;
3. int[] []a,b;
4. **int[] []a,[]b; (CE)**

Note: If we want to specify the dimension before the variable it is possible only for the first variable

ARRAY CONSTRUCTION:

- Every array in java is an object hence we can create by using new operator.
  - **Ex:** int[] a=new int[3];
- At the time of construction compulsory we should specify the size otherwise we will get Compilation Error.
  - int[] a= new int[]; **Compilation error**
  - int[] a=new int[3];
- It is legal to have an array of size 0 in java
  - **Ex:** int[] a=new int[0];
- If we are specifying array size as -ve int value we will get runtime exception saying
  Negativeruntimeexception
  - **Ex:** int[] a=new int[-6]
- To specify array size the allowed datatypes are byte, short, int, char. If we use other data type we will get compilation error.

FIND THE VALID ARRAY STATEMENT:

1. int[] a = new int[‘a’];
2. Byte b=10;
3. int[] a = new int[b];
4. int[] a = new int[10l];
5. int[] a= new int[10.5];

FIND THE VALID ARRAY STATEMENT:

- int[] a = new int[‘a’];
- Byte b=10;
- int[] a = new int[b];
- int[] a = new int[10l];
- int[] a= new int[10.5];
- To specify array size the allowed datatypes are byte, short, int, char. If we use other data type we will get compilation error.
- Note: the max allowed array size in java is 2147483647 (max value of int datatype)

ARRAY INITIALIZATION:

- **Whenever we are creating an array automatically every element is initialized with default values.**

Predict the output:

```
import java.util.*;
public class Main
{

  public static void main (String[]args)
  {
    int[] a= new int[3];
    System.out.println(a);
    System.out.println(a[0]);
  }

}
```

Output:

```
[I@2a139a55
0
```

ASSIGNING VALUES IN AN ARRAY:

- Once we created an array every element by default initialized with default values if we are not satisfy with those values then **we can override with our customized value.**
- int[] a=new int[5];
- a[0]=10;
- a[1]=20;
- a[50]=50;
- a[10.5]=30;

ASSIGNING VALUES IN AN ARRAY:

- Once we created an array every element by default initialized with default values if we are not satisfy with those values then **we can override with our customized value.**
- int[] a=new int[5];
- a[0]=10;
- a[1]=20;
- a[50]=50; **(run time error)**
- a[10.5]=30; **(compilation error)**

ARRAY DECLARATION, CONSTRUCTION AND INITIALIZATION IN A SINGLE LINE:

- We can declare construct and initialize an array into a single line
- **int[] a;**

a=new int[3];

a[0]=10;

a[1]=20;

a[2]=30;

a[3]=40;

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

TYPES OF ARRAY

- **Single dimensional array**
- **Multi dimensional array**

SINGLE DIMENSIONAL ARRAY

```
class Array
{
   public static void main(String args[])
   {
      int a[]=new int[5];      //declaration and instantiation
      a[0]=5;       //initialization
      a[1]=10;
      a[2]=15;
      a[3]=20;
      a[4]=25;
      for(int i=0; i<a.length; i++)     //a.length is used to find the size
      {
         System.out.println(a[i]);
      }
   }
}
```

SINGLE DIMENSIONAL ARRAY

```
class Array
{
   public static void main(String args[])
   {
      int a[]=new int[5];      //declaration and instantiation
      a[0]=5;       //initialization
      a[1]=10;
      a[2]=15;
      a[3]=20;
      a[4]=25;
      for(int i=0; i<a.length; i++)     //a.length is used to find the size
      {
         System.out.println(a[i]);
      }
   }
}
```

Output

5
10
15
20
25

SINGLE DIMENSIONAL ARRAY

```
import java.util.*;
class Array8
{
   public static void main(String args[])
   {
        int length;
        Scanner S=new Scanner(System.in);
        System.out.print("Enter Length of Array8: ");
        length=S.nextInt();
        int a[]=new int[length];
        System.out.print("Enter the elements of Array8: ");
        for(int i=0; i<length; i++)
        {
           a[i] = S.nextInt();
        }
        System.out.print("Elements of Array8 are: ");
        for(int i=0; i<length; i++)
        {
           System.out.print(a[i] + "  ");
        }
   }
}
```

SINGLE DIMENSIONAL ARRAY

```
import java.util.*;
class Array8
{
   public static void main(String args[])
   {
        int length;
        Scanner S=new Scanner(System.in);
        System.out.print("Enter Length of Array8: ");
        length=S.nextInt();
        int a[]=new int[length];
        System.out.print("Enter the elements of Array8: ");
        for(int i=0; i<length; i++)
        {
           a[i] = S.nextInt();
        }
        System.out.print("Elements of Array8 are: ");
        for(int i=0; i<length; i++)
        {
           System.out.print(a[i] + "  ");
        }
   }
}
```

Output
Enter Length of Array8:
Enter the elements of Array8:
Elements of Array8 are:

SINGLE DIMENSIONAL ARRAY (CONT.…)

Creating an array of objects

```
class Array1
{
    int ID;
    String Name;
    Array1(int Identity, String Full_name)
    {
        this.ID = Identity;
        this.Name = Full_name;
    }
    public static void main (String[] args)
    {
        Array1 [] my_arr;
        my_arr = new Array1[5];
        my_arr[0] = new Array1(101,"Tanuj");
        my_arr[1] = new Array1(202,"Anwesh");
        my_arr[2] = new Array1(303,"Tarun");
        for (int i = 0; i < my_arr.length; i++)
        {
            System.out.println("Values of my_arr " + i + " : " + my_arr[i].ID +" "+ my_arr[i].Name);
        }
    }
}
```

SINGLE DIMENSIONAL ARRAY (CONT.…)

Creating an array of objects

```
class Array1
{
    int ID;
    String Name;
    Array1(int Identity, String Full_name)
    {
        this.ID = Identity;
        this.Name = Full_name;
    }
    public static void main (String[] args)
    {
        Array1 [] my_arr;
        my_arr = new Array1[5];
        my_arr[0] = new Array1(101,"Tanuj");
        my_arr[1] = new Array1(202,"Anwesh");
        my_arr[2] = new Array1(303,"Tarun");
        for (int i = 0; i < my_arr.length; i++)
        {
            System.out.println("Values of my_arr " + i + " : " + my_arr[i].ID +" "+ my_arr[i].Name);
        }
    }
}
```

Output
Values of my_arr 0 : 101 Tanuj
Values of my_arr 1 : 202 Anwesh
Values of my_arr 2 : 303 Tarun
Exception

SINGLE DIMENSIONAL ARRAY (CONT.…)

Passing an array to a method

```
class Array2
{
   static void max(int my_arr[])
   {
      int max=my_arr[0];
      for(int i=1; i<my_arr.length; i++)
      {
         if(my_arr[i]>max)
         {
            max=my_arr[i];
         }
      }
      System.out.println(max);
   }
   public static void main(String args[])
   {
      int X[]={45, 87, 76, 100, 56, 301};
      max(X);      //passing array to the method max
   }
}
```

SINGLE DIMENSIONAL ARRAY (CONT.…)

Passing an array to a method

```
class Array2
{
   static void max(int my_arr[])
   {
      int max=my_arr[0];
      for(int i=1; i<my_arr.length; i++)
      {
         if(my_arr[i]>max)
         {
            max=my_arr[i];
         }
      }
      System.out.println(max);
   }
   public static void main(String args[])
   {
      int X[]={45, 87, 76, 100, 56, 301};
      max(X);      //passing array to the method max
   }
}
```

Output
301

SINGLE DIMENSIONAL ARRAY (CONT.…)

Returning an array from a method

```
class Array3
{
   static int[] hello()
   {
      return new int[]{100, 200, 400, 800, 1600};
   }
   public static void main(String args[])
   {
     int my_arr[]=hello();       //calling method
     for(int i=0; i<my_arr.length; i++)
     {
        System.out.println(my_arr[i]);
     }
   }
}
```

SINGLE DIMENSIONAL ARRAY (CONT.…)

Returning an array from a method

```
class Array3
{
   static int[] hello()
   {
      return new int[]{100, 200, 400, 800, 1600};
   }
   public static void main(String args[])
   {
     int my_arr[]=hello();       //calling method
     for(int i=0; i<my_arr.length; i++)
     {
        System.out.println(my_arr[i]);
     }
   }
}
```

Output

100
200
400
800
1600

MULTI DIMENSIONAL ARRAY

- A multidimensional array is an array that contains one or more arrays.
- In multidimensional array, element of the array holds the reference of another array .
- A multidimensional array is created by using one set of square brackets “[]” per dimension.
- It is mostly used.

MULTI DIMENSIONAL ARRAY (CONT.…)

- To declare a multidimensional array variable, specify each additional index using another set of square brackets.

```
int twoDimen[ ][ ] = new int[4][5];
int[][] my_arr = { {1, 2, 3, 4}, {5, 6, 7} };
```

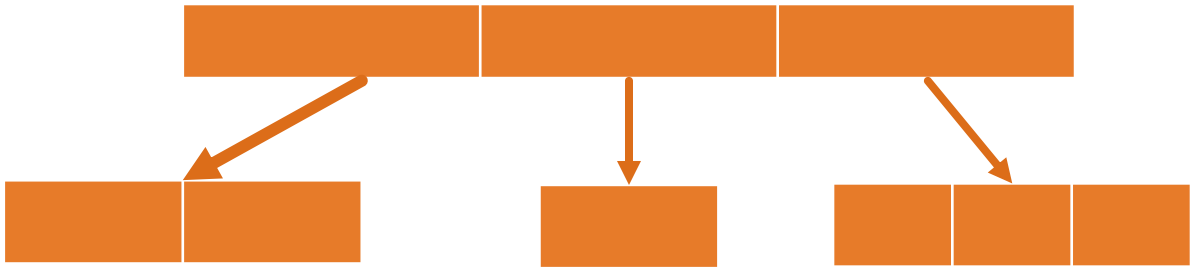
MULTI DIMENSIONAL ARRAY

- In java multi dimensional arrays are not implemented in matrix form. They are implemented by using array of array concept.
- The main advantage of this approach is memory utilization will be improved.
- **Ex1:** int[][] a=new int[3][];

a[0]= new int[2];

a[1]= new int[1];

a[2]= new int[3];



MULTI DIMENSIONAL ARRAY

- **Ex2:**
- int[][][] a=new int[2][][];

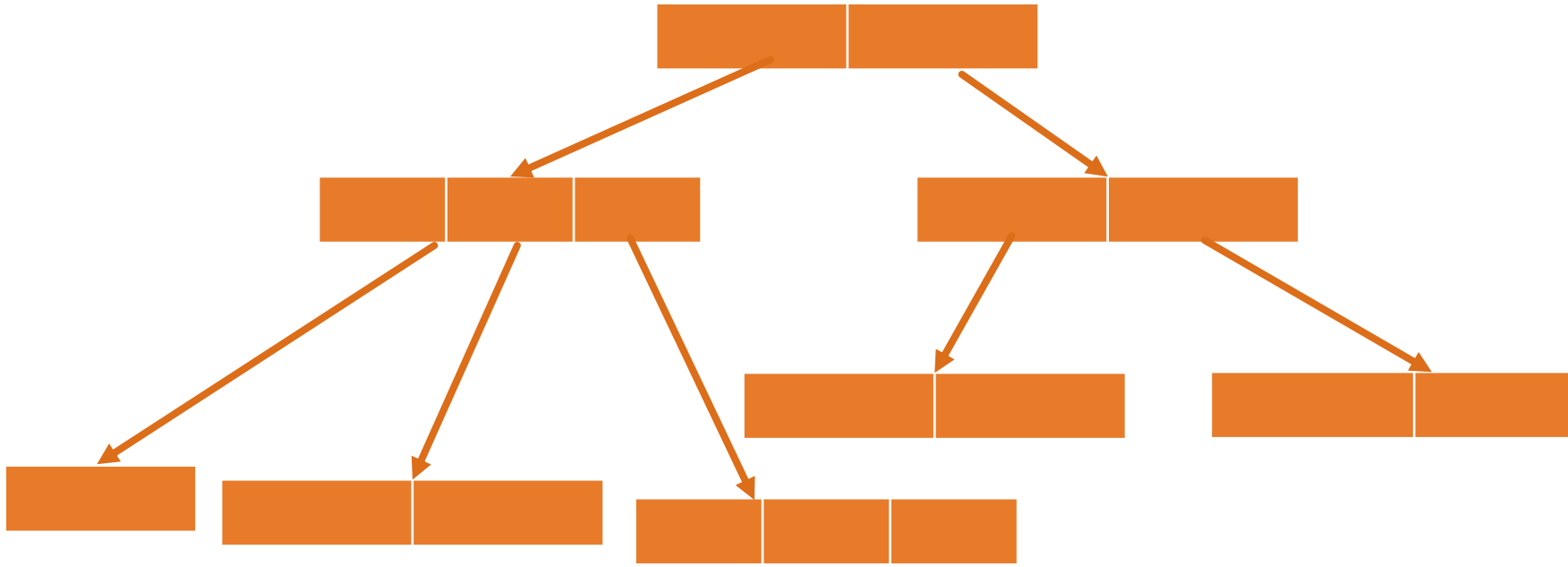
a[0]= new int[3][];

a[0][0]= new int[1];

a[0][1]= new int[2];

a[0][2]= new int[3];

a[1]= new int[2][2];



MULTI DIMENSIONAL ARRAY (CONT.…)

```
class Array5
{
  public static void main(String args[])
  {
    int my_arr[][] = {{34, 87, 39},{31, 65, 12},{27, 64, 29}};
    for (int i=0; i<=2 ; i++)
    {
      for (int j=0; j <=2 ; j++)
      {
        System.out.print(my_arr[i][j] + " ");
      }
      System.out.println();
    }
  }
}
```

MULTI DIMENSIONAL ARRAY (CONT.…)

```
class Array5
{
  public static void main(String args[])
  {
    int my_arr[][] = {{34, 87, 39},{31, 65, 12},{27, 64, 29}};
    for (int i=0; i<=2 ; i++)
    {
      for (int j=0; j <=2 ; j++)
      {
        System.out.print(my_arr[i][j] + " ");
      }
      System.out.println();
    }
  }
}
```

Output

34 87 39

31 65 12

27 64 29

MULTI DIMENSIONAL ARRAY (CONT.…)

```
public class Array6
{
  public static void main(String args[])
  {
    int a[][]={{1, 3, 5}, {7, 9, 11}};
    int b[][]={{2, 4, 6}, {8, 10, 12}};
    int c[][]=new int[2][3];
    for(int i=0; i<2; i++)
    {
      for(int j=0; j<3; j++)
      {
         c[i][j]=a[i][j] + b[i][j];
         System.out.print(c[i][j] + " ");
      }
      System.out.println();
    }
  }
}
```

Output
3 7 11
15 19 23

MULTI DIMENSIONAL ARRAY (CONT.…)

Write a Java program to multiply two matrices:

$$A = \begin{matrix} 2 & 4 & 6 \\ 8 & 10 & 12 \\ 14 & 16 & 18 \end{matrix}$$

$$B = \begin{matrix} 1 & 3 & 5 \\ 7 & 9 & 11 \\ 13 & 15 & 17 \end{matrix}$$

LENGTH OF AN ARRAY:

- **It is a final variable applicable only for arrays**
- **It represent the size of array**
- **Ex. Int[] a=new int[10];**
- **Sop(a.length); //output : 10**
- **Sop(a.length()); // Compilation Error**

THANK YOU
?