DAY-17

------

1. VALUE TYPE ASSIGNMENT

2. REFERNCE TYPE ASSIGNMENT

1. VALUE TYPE ASSIGNMENT: Whenever value of one variable is assigned to another variable is called as value type assignment.

example:

--------

Demo.java

---------

class Demo

{

public static void main(String[] args)

{

int x = 10;

int y;

y = x;

System.out.println(x);

System.out.println(y);

}

}

output:

-------

10

10

2. REFERENCE TYPE ASSIGNMENT : Whenever the address of the one variable is assigned to another variable then it is called as reference type assignment.

After the assignment all the ref variable would be pointing to the same object.

Modification done by any one of the reference variable would affect the same object.

NOTE: In java one object can have multiple reference variables.

example:

--------

Demo.java

---------

class Demo

{

String name;

int roll\_no;

public static void main(String[] args)

{

Demo d1 = new Demo();

d1.name = "sagar";

d1.roll\_no = 24;

System.out.println(d1.name);

System.out.println(d1.roll\_no);

Demo d2;

d2 = d1;

d1.name = "sandesh";

d1.roll\_no = 25;

System.out.println(d2.name);

System.out.println(d2.roll\_no);

System.out.println(d1.name);

System.out.println(d1.roll\_no);

}

}

output:

-------

sagar

24

sandesh

25

sandesh

25

---------------------------------------------------------------------------------------------------------------------------------------------------------------

ARRAYS

-------

Whenver we try to store large amount of data in a variable we will face readability issue.

In order to store small amount data we can use variables but if we want to store large amount of data then we must use ARRAYS.

--> An array is an indexed collection of fixed number of homogeneous data elements.

--> The main disadvantage of array is we can present only homogeneous type of data.

--> The main advantage of array is we can represent multiple values under the same variable.Hence we can achive readability.

--> Once the array is created with some size we can not alter that during execution.

--> We can use concept called as COLLECTIONS to slove the above mentioned problem.

TYPES OF ARRAYS :

-----------------

1. SINGLE DIMENSION ARRAY

2. MULTI DIMENSION ARRAY

1. SINGLE DIMENSION ARRAY [1-D]

-------------------------------

In this type the data can be stored in the single row/column.

SYNTAX:

-------

There are 3 steps in order to create 1d array.

1. Declaration of a array variable. eg: int a[];

2. Creation of array object. eg: int a = new int[5];

3. Initilization of a array. eg: a[0] = 10;

a[1] = 20;

note: int a[] = new int[5];

Single line(declare create initilize

-------------------------------------

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

NOTE: Array is an obejct since it is created using new keyword. The Super class of array is Object.

WHERE DOES THE MEMORY ALLOCATED FOR ARRAYS?

--> The arrays are created inside the object hence the memeory is allcated in heap memeory segment.

Note: When ever array is created the default values will be added based on the datatype.

example:

--------

int a[] = new int[5] --> 0 0 0 0 0

float a[] = new float[5] --> 0.0 0.0 0.0 0.0 0.0

ARRAY DECLARATION:

-----------------

The declaration of the array can be done in nay one of the way:

1. int a[];

2. int []a;

3. int[] a; -->{recommended}

example:1

---------

Demo1.java

----------

class Demo1

{

public static void main(String[] args)

{

int a[];

a = new int[5];

a[0] = 10;

a[1] = 20;

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

System.out.println(a[1]);

System.out.println(a[2]);

System.out.println(a[3]);

System.out.println(a[4]);

//System.out.println(a[5]); --> error: ArrayIndexOutOfBoundsException: 5

}

}

output:

-------

10

20

0

0

0

example 2:

----------

Demo1.java

----------

class Demo1

{

public static void main(String[] args)

{

int a[] = new int[5];

a[0] = 10;

a[1] = 20;

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

System.out.println(a[1]);

System.out.println(a[2]);

System.out.println(a[3]);

System.out.println(a[4]);

//System.out.println(a[5]);

}

}

output:

-------

10

20

0

0

0

example 3:

----------

Demo1.java

class Demo1

{

public static void main(String[] args)

{

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

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

System.out.println(a[1]);

System.out.println(a[2]);

System.out.println(a[3]);

System.out.println(a[4]);

//System.out.println(a[5]);

}

}

output:

-------

10

20

30

40

50

---------------------------------------------------------------------------

Class name for different array types:

ARRAY TYPE CLASS NAME

---------- -----------

--> int[] --> [I

--> byte[] --> [B

--> short[] --> [S

--> long[] --> [J

--> float[] --> [F

--> double[] --> [D

--> char[] --> [C

example:

--------

Demo1.java

-----------

class Demo1

{

public static void main(String[] args)

{

//float a[] = new float[5]; --> [F

//int a[] = new int[5]; --> [I

short s[] = new short[5]; --> [S

System.out.println(s.getClass());

}

}

NOTE: At the array declaration Size should not be specified. Size should be mentioned only during creation of aaray.

example:

--------

int a[5]; --> invalid

int a[]; --> valid

It is valid to create array with size zero

--> int a[] = new int[0]; --> valid.

An array size can not be mentioned as negative value. If we do so during compilation we will not get any problem but during execution we will get error

--> int a[] =new int[-5]; --> error NegativeArraySizeException