DAY-16

------

STATIC VARIABLE:

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

1. WHAT IS STATIC VARIABLE ?

--> The value of a variable does not change from object to object such variables are called as STATIC VARIBALES.

2.WHERE THE STATIC VARIABLES ARE DECLARED ?

--> Static variables are declared within the class but outside the methods or block.

example:

--------

class People

{

String name;

int addhar\_no;

static String nationality;

}

NOTE: For all the objects only one copy of static variable is created and it is shared by every object.

3. WHERE DOES THE MEMORY ALLOCATED FOR STATIC VARIABLES ?

--> Static variables always present in method area [static memory segment] inside the memory.

4. WHAT IS THE SCOPE OF THE STATIC VARIABLE ?

--> Scope refers to the memory allocation and deallocation

For the static variable memory will be allocated during loading of class file and deallocated while unloading the class file.

Hence scope of the static variable is same as the .class file.

5. HOW WE CAN ACESS THE STATIC VARIABLE ?

--> We can access the static variable using:

directly we can use within the static method

using the object reference

using class name {recommeded}

example:

--------

People.java

-----------

class People

{

String name;

int addhar\_no;

static String nationality="Indian";

public static void main(String[] args)

{

People p1 = new People();

System.out.println(nationality); // directly we can access

System.out.println(p1.nationality); // using object reference

System.out.println(People.nationality); // using class name [recommended]

}

//System.out.println(nationality);

}

output:

-------

Indian

Indian

Indian

6.JVM always gives default values for the static variables:

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

example:

--------

Demo2.java

----------

class Demo2

{

static byte b;

static short s;

static int i;

static long l;

static float f;

static double d;

static boolean bo;

static char ch;

static String st;

public static void main(String[] args)

{

Demo2 d1 = new Demo2();

System.out.println(d1.b); --> 0

System.out.println(d1.s);` --> 0

System.out.println(d1.i); --> 0

System.out.println(d1.l); --> 0

System.out.println(d1.f); --> 0.0

System.out.println(d1.d); --> 0.0

System.out.println(d1.bo); --> false

System.out.println(d1.ch); --> space

System.out.println(d1.st); --> null

}

}

NOTE: If the values of static variable is modified then it will be reflected on all the objects

example:

--------

Demo.java

---------

class Demo

{

static int x = 25;

int y = 50;

public static void main(String[] args)

{

Demo d1 = new Demo();

System.out.println(d1.x); // 25

System.out.println(d1.y); // 50

d1.x = 143;

d1.y = 420;

System.out.println(d1.x); // 143

System.out.println(d1.y); // 420

Demo d2 = new Demo();

System.out.println(d2.x); // 143

System.out.println(d2.y); // 50

}

}

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

LOCAL VARIABLE:

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

1. WHAT IS LOCAL VARIABLE ?

--> Local Variables are temporory variables which are used by programmers to meet temporory requierments.

2. WHERE THE LOCAL VARIABLES ARE DECLARED ?

--> Local Variables are always declared within the method or block.

example:

--------

Class Demo

{

p s v m(String []args)

{

int x =10;

for(int i=0;i<10;i++)

{

s.o.p(i);

}

}

}

3. WHERE DOES LOCAL VARIABLES ARE LOCATED IN THE MEMORY ?

--> Local Variables are always present in stack memory hence it is also called as stack variable.

4. WHAT IS THE SCOPE OF THE LOCAL VARIABLE ?

--> Scope refers to memeory allocation and deallocation.

memory will be allocated when ever control enters the method/block.

memory will be deallocated when ever control leaves the method/block.

Hence the scope of Local variables is same as the method scope.

example:

class Demo

{

public static void main(String[] args)

{

int x = 10;

System.out.println(x); // valid

for (int i=0;i<10 ;i++ )

{

System.out.println(i); // valid

System.out.println(x); // valid

}

//System.out.println(i); // error[invalid]

}

}

5. HOW WE CAN ACCESS THE LOCAL VARIABLES ?

--> Local Variables can be accessed only within the method/block.Outside the method we can not access.

example: refer above example

6.DOES THE JVM PROVIDE DEFAULT VALUES FOR LOCAL VARIBALES?

-->NO,Programmer must initilize the values to the local variable.

example:1

--------

Demo.java

---------

class Demo

{

public static void main(String[] args)

{

int x;

System.out.println("java"); --> if local variables are not used then we wont get compilation error

}

}

output:

-------

java

example:2

---------

Demo.java

---------

class Demo

{

public static void main(String[] args)

{

int x;

System.out.println("java");

System.out.println(x); --> if we try to use local variables with out initilization then we will get compilation

error

}

}

output:

-------

error

example:3

---------

NOTE: jvm will not provide default values to the local variables hence it is recommended to the programmer to give default values

Demo.java

---------

class Demo

{

public static void main(String[] args)

{

int x = 0;

System.out.println("java");

System.out.println(x);

}

}

ouput:

------

java

0