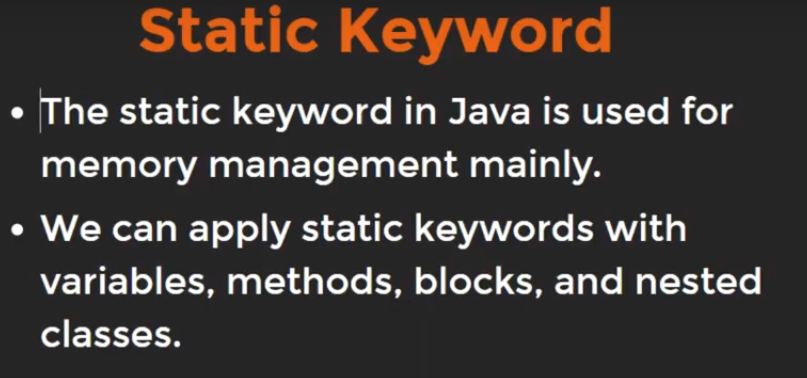
**Static keyword & Static Variable**

****

Static Variables doesn’t Take dynamic value.

**Note:**

* Since, int variable takes 4 byte in memory.
* If we initialize an int variable under a class, then it will not take any memory in the RAM. Because As we define class it doesn’t take any memory until we not create its instance i.e. an object of that class. As we create an object, all variables associating with the class start taking space in the memory. Like 4 byte for int variables.
* So, as many times we create instances of that class, it will take space for all variable in the memory as the same number of times.
* But if we declare variable with static keyword, it will take space only once for all variables in the memory irrespective as many times we create instance of that class.
* Static keyword makes variable to allocate memory only once and use multiple times.
* A without static variable treat as different variable for different objects whereas a static variable treat as same variable for different objects.

**Example: (Variable without static keyword)**

public class Main {

// main Method

public static void main(String[] args) {

Person per1 = new Person();

Person per2 = new Person();

per1.getCounter();

per2.getCounter();

}

}

class Person {

int counter = 0;

void getCounter(){

counter++;

System.out.println(counter);

}

}

**Output:-**1  
1

**Example: (Variable with static keyword)**

public class Main {

// main Method

public static void main(String[] args) {

Person per1 = new Person();

Person per2 = new Person();

per1.getCounter();

**//due to static variable, counter will increase from 1 not from 0 i.e. starting value.**

per2.getCounter();

}

}

class Person {

**//counter get changed from the same memory location from where it was previously changed.**

static int counter = 0;

void getCounter(){

counter++;

System.out.println(counter);

}

}

**Output:-**1  
2