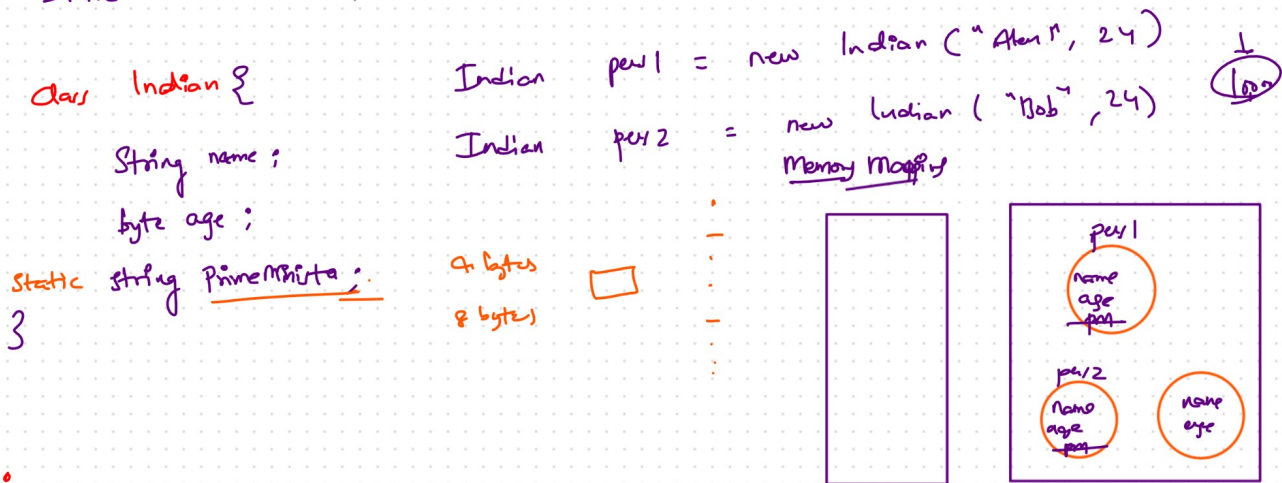


Static keyword

Static variable:

- We can make a variable as static when it is common for all the instances of that particular class
- static variables are also called as "class variables"
 - ↳ one copy for whole class
- the memory for the static variable will be allocated inside the method area during the time [class loading]
- static variable improves memory management



Note:

- Before v. 1.8, static variables are get created in special memory area called "permanent Generation" in method area
- After 1.8, it gets removed & stored in method area itself

Notes:

→ Static variables can be accessed with the help of two types

- (1) object Reference
- (2) class Name

```
Indian person1 = new Indian();  
Indian person2 = new Indian();  
person1.name = "Alex";  
person1.age = 24;  
person2.name = "Bob";  
person2.age = 25;  
→ person1.primeMinister = "Modi";
```

```
System.out.println(person3.primeMinister);  
→ Indian.primeMinister = "Modiji";  
System.out.println(person1.primeMinister);  
System.out.println(person2.primeMinister);
```

Blocks in Java

{ }

→ there are 2 types of blocks in Java

- 1. Instance Block
- 2. Static Block

① Instance Block

```
{  
_____  
_____  
_____  
_____  
}
```

②

Static Block

```
static {  
_____  
_____  
_____  
}
```

① Instance Block

- Instance blocks are written inside the class & outside the method
- Instance blocks are executed every time when object is created
- If we create 10 objects then ten times the block gets executed

② Static Block

- (1) Static blocks are declared inside the class & outside the method
- (2) Static blocks are executed during the time of class loading & it gets executed only once


```

1 class AlphaBlocks{
2 {
3     System.out.println("Instance Block - 1");
4 }
5 static {
6     System.out.println("Static Block -1");
7 }
8
9 {
10    System.out.println("Instance Block -2 ");
11 }
12
13 static {
14     System.out.println("Static Block -2");
15 }
16 }
17 public class BlocksProg {
18 {
19     System.out.println("Main instance Block");
20 }
21 public static void main(String[] args) {
22     AlphaBlocks obj1 = new AlphaBlocks();
23     AlphaBlocks obj2 = new AlphaBlocks();
24 }
25 static {
26     System.out.println("Main static Block");
27 }
28 }

```

Note :

1. Before loading the class [main], it checks for any static variable & then for static blocks, these are executed if no any variable/block then it checks for static method, so that is why the main method gets executed & reason behind to keep main() as "static"
2. In the above prog, when ever the obj of type AlphaBlocks is created it loads the class Alpha in the method Area [Application loader loads it] & then it checks for static variable, if not (or) then proceeds to static block, so static block are gets executed first before object creation
3. After static creation is done object checks for instance variable which are loaded into heap Area & it proceeds to instance blocks & then executes it.
4. If the same type of another object is created (obj2) it doesn't load once again the class of that type into method Area & it doesn't executes the static variables followed by static blocks...