



Fundamentals of Object Oriented Programming

CSN- 103

Dr. R. Balasubramanian

Associate Professor

Department of Computer Science and Engineering

Indian Institute of Technology Roorkee

Roorkee 247 667

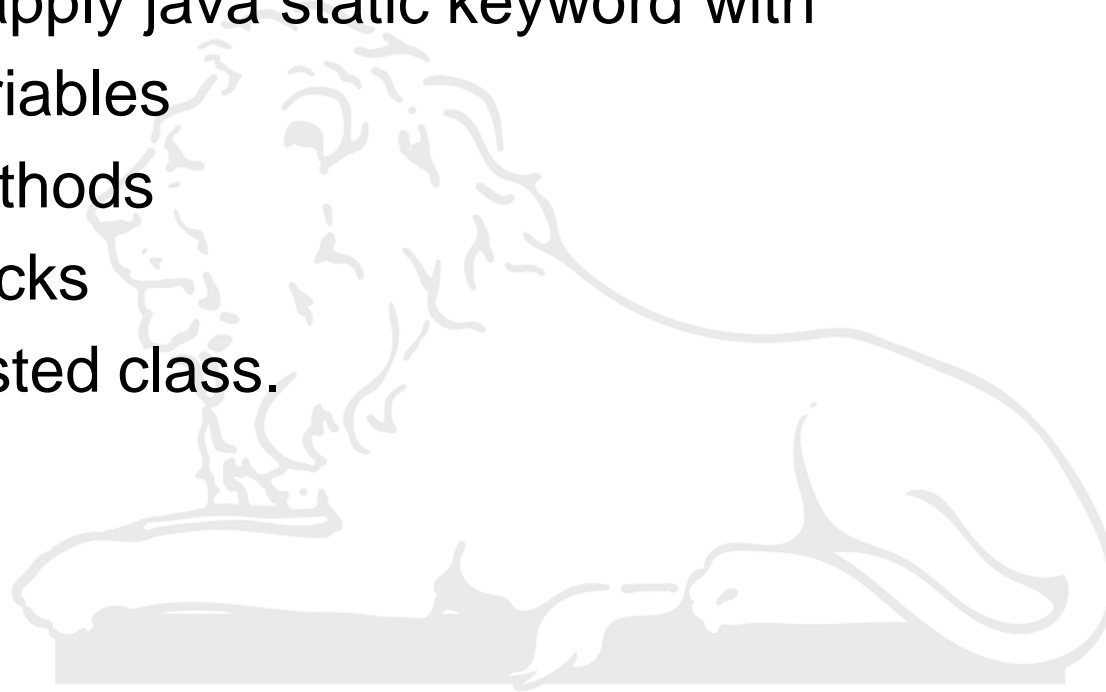
balarfcs@iitr.ac.in

<https://sites.google.com/site/balaiiitr/>



JAVA static word

- The **static keyword** in java is used for memory management mainly.
- We can apply java static keyword with
 - Variables
 - methods
 - blocks
 - nested class.





Java static variable

- If you declare any variable as static, it is known as static variable.
- The static variable can be used to refer the common property of all objects (that is not unique for each object) e.g. company name of employees, college name of students etc.
- The static variable gets memory only once in class area at the time of class loading.
- **Advantage of static variable**
 - It makes your program **memory efficient** (i.e. it saves memory).



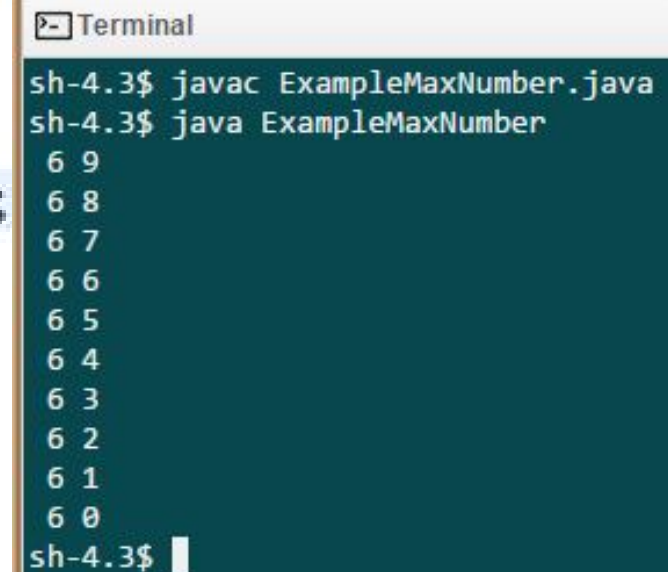
Static variable

```
1 public class ExampleMaxNumber{
2     static int count=10;
3     static int i=5;
4     public static void main(String[] args) {
5         while (count-->0)
6             {checkStatic();}
7     }
8
9     public static void checkStatic() {
10         i++;
11         System.out.print(" " +i);
12         System.out.println(" " +count);
13     }
14 }
15
```

Terminal

```
sh-4.3$ javac ExampleMaxNumber.java
sh-4.3$ java ExampleMaxNumber
6 9
7 8
8 7
9 6
10 5
11 4
12 3
13 2
14 1
15 0
sh-4.3$
```

```
1 public class ExampleMaxNumber{
2     static int count=10;
3     public static void main(String[] args) {
4         while (count-->0)
5             {checkStatic();}
6     }
7
8     public static void checkStatic() {
9         int i=5;
10        i++;
11        System.out.print(" " +i);
12        System.out.println(" " +count);
13    }
14 }
15
```



```
Terminal
sh-4.3$ javac ExampleMaxNumber.java
sh-4.3$ java ExampleMaxNumber
6 9
6 8
6 7
6 6
6 5
6 4
6 3
6 2
6 1
6 0
sh-4.3$
```



```
1 class Counter{
2   int count=0;//will get memory when instance is created
3
4   Counter(){
5     count++;
6     System.out.println(count);
7   }
8
9   public static void main(String args[]){
10
11     Counter c1=new Counter();
12     Counter c2=new Counter();
13     Counter c3=new Counter();
14
15   }
16 }
```

Terminal

```
sh-4.3$ javac Counter.java
sh-4.3$ java Counter
1
1
1
sh-4.3$
```



```
1 class Counter2{
2     static int count=0;//will get memory only once and retain its value
3
4     Counter2(){
5         count++;
6         System.out.println(count);
7     }
8
9     public static void main(String args[]){
10
11         Counter2 c1=new Counter2();
12         Counter2 c2=new Counter2();
13         Counter2 c3=new Counter2();
14
15     }
16 }
```

Terminal

```
sh-4.3$ javac Counter2.java
sh-4.3$ java Counter2
1
2
3
sh-4.3$
```



Addition of Two distances

```
1 class distance{
2     int feet;
3     int inches;
4
5     void setDistance(int x , int y)
6     {
7         feet=x;
8         inches=y;
9     }
10    void displaydistance()
11    {
12        System.out.println(feet+" feet" + " " +inches+" inchess");
13    }
14
15    int getFeet()
16    {
17        return feet;
18    }
19
20    int getInches()
21    {
22        return inches;
23    }
24 }
25
```

distance ()
{ }



```
26 class Executedistance1 {
27
28     public static void main(String[] args) {
29         distance d1, d2, d3;
30         d1=new distance();
31         d2=new distance();
32         d3=new distance();
33         // System.out.println("the first distance is :");
34         d1.setDistance(10,9);
35         d1.displaydistance();
36         d2.setDistance(9,10);
37         d2.displaydistance();
38         // System.out.println("the second distance is :");
39         int ft=d1.getFeet()+d2.getFeet();
40         int inc=d1.getInches()+d2.getInches();
41         if(inc>=12)
42         {
43             ft++;
44             inc=inc-12;
45         }
46         d3.setDistance(ft,inc);
47         d3.displaydistance();
48     }
49
50 }
51
```

Terminal

```
sh-4.3$ javac Executedistance1.java
sh-4.3$ java Executedistance1
10 feet 9 inchess
9 feet 10 inchess
20 feet 7 inchess
sh-4.3$
```





Objects as Arguments

```
1 class distance{
2     int feet;
3     int inches;
4     distance()
5     { }
6     distance(int x , int y)
7     {
8         feet=x;
9         inches=y;
10    }
11    void displaydistance()
12    {
13        System.out.println(feet+" feet" + " " +inches+" inchess");
14    }
15    void addDistance(distance one,distance two)
16    {
17        feet=one.feet+two.feet;
18        inches=one.inches+two.inches;
19        if(inches>=12)
20        {
21            feet++;
22            inches=inches-12;
23        }
24    }
25 }
26 }//distance type created
27
```



```
28 class Executedistance2{
29
30     public static void main(String[] args) {
31         distance d1=new distance(10,9);
32         System.out.println("the first distance is :");
33         d1.displaydistance();
34         distance d2=new distance(9,10);
35         System.out.println("the second distance is :");
36         d2.displaydistance();
37         distance d3=new distance();
38         d3.addDistance(d1,d2);
39         System.out.println("the sum of their distance is :");
40         d3.displaydistance();
41
42     }
43
44 }
45
```

Terminal

```
sh-4.3$ javac Executedistance2.java
sh-4.3$ java Executedistance2
the first distance is :
10 feet 9 inchess
the second distance is :
9 feet 10 inchess
the sum of their distance is :
20 feet 7 inchess
sh-4.3$
```
