```
public static void main(String[] args)
{
   String[] s = new String[2];
   int i =0;
   for(String s1 : s)
   {
      s[i].concat("Element "+i);
      i++;
   }
   for(i=0;i<s.length;i++)
   {
      System.out.println(s[i]);
   }
}
```

What is the result?

A) Element 0
   Element 1

B) null Element 0
   null Element 1

C) null
   null

D) NullPointerException

Q8. Consider the code

```
class Test
{
      public static void main(String[] args)
      {
            int[][] n = new int[1][3];
            for(int i =0; i< n.length; i++)
            {
                  for (int j=0;j>n[i].length ;j++ )
                  {
                        num[i][j]=10;
                  }
            }
      }
}
```

```
class Student
{
   String name;
   public Student(String name)
   {
      this.name=name;
   }
}
public class Test
{
   public static void main(String[] args)
   {
      Student[] students = new Student[3];
      students[1]= new Student("Durga");
      students[2]= new Student("Ravi");
      for(Student s : students)
      {
         System.out.println(s.name);
      }
   }
```

What is the result?

A) Durga
   Ravi

B) null
   Durga
   Ravi

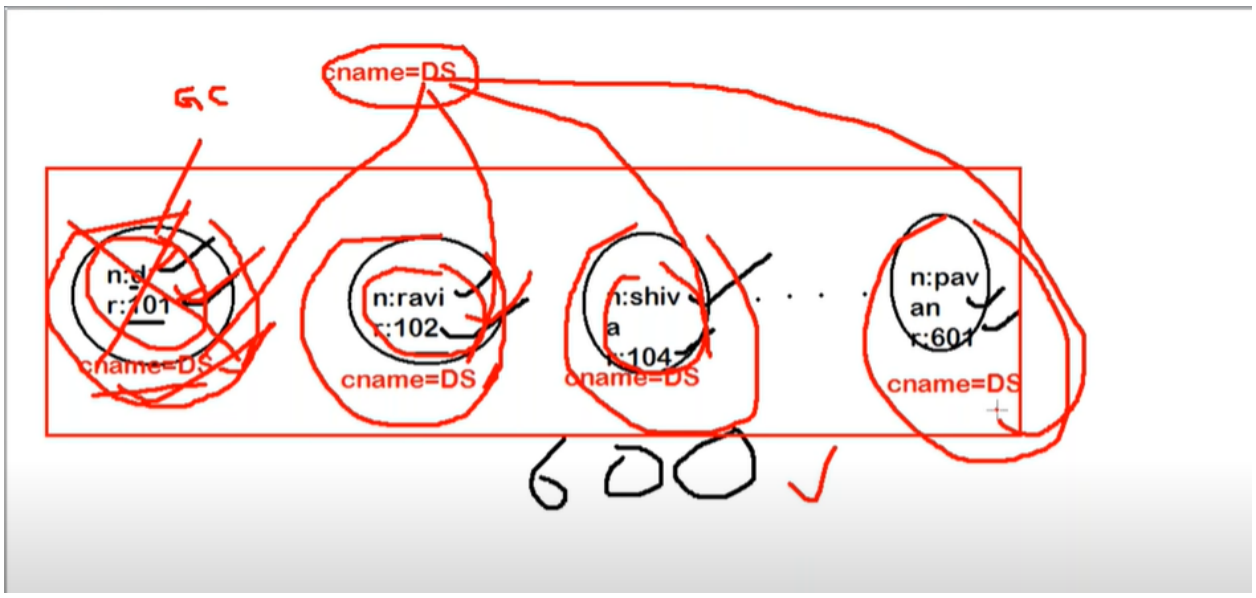
C) Compilation Fails

D) ArrayIndexOutOFBoundsException

E) NullPointerException

```
public class Test
{
    int x=10;
    public static void main(String[] args)
    {
        m1();
    }
    public void m1()
    {
        System.out.println(x);
    }
}
```

```java
public class Test
{
    int x;
    public static void main(String[] args)
    {
        Test t = new Test();
        System.out.println(t.x);
    }
}
```



Static variable creates at the start of class loading and destroy at the time of class unloading.

Flow of java test

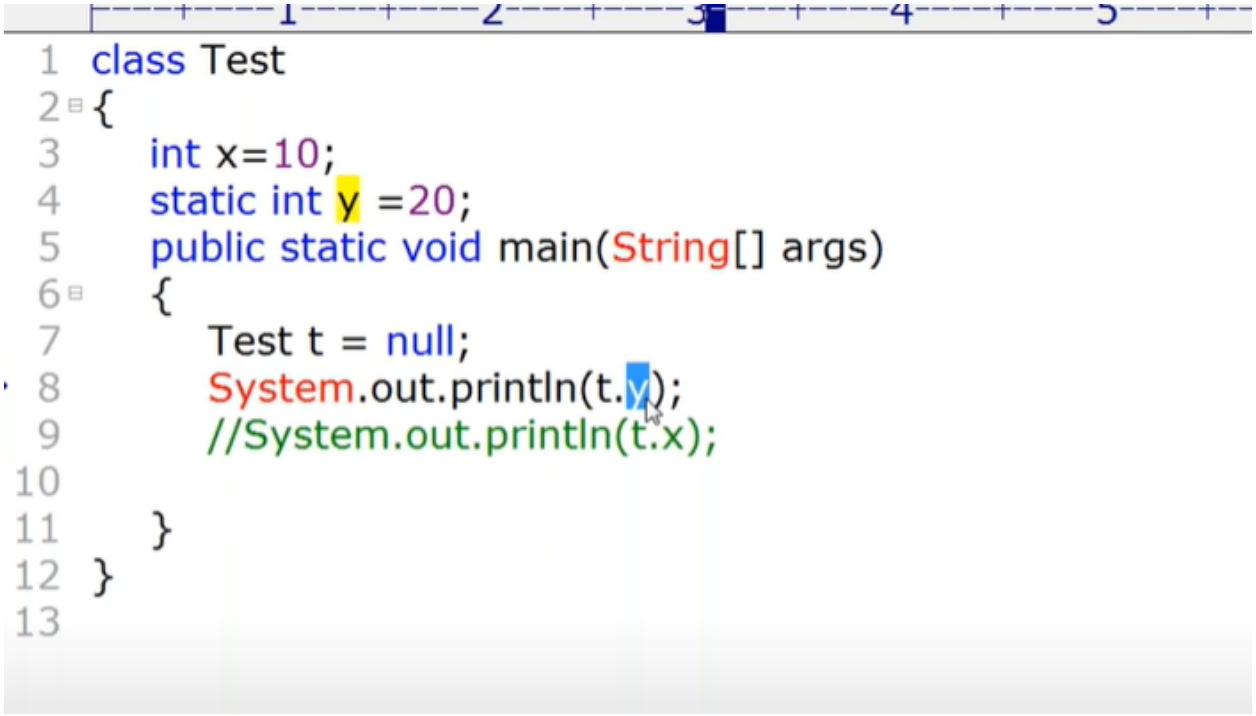
```
D:\durgaclasses>java Test

1. JVM will be started
2. Create and Start Main Thread by JVM
3. Main Thread will search for Test.class==>(NoClassDefFoundError)
4. Main Thread will load Test.class
5. Execute main method
6. unload Test.class file
7. Terminate Main Thread
8. JVM will be shutdown
```

```
D:\durgaclasses>java Test

1. JVM will be started
2. Create and Start Main Thread by JVM
3. Main Thread will search for Test.class==>(NoClassDefFoundError)
4. Main Thread will load Test.class(here static variables will be created)
5. Execute main method
6. unload Test.class file(here static variables will be destroyed)
7. Terminate Main Thread
8. JVM will be shutdown
```

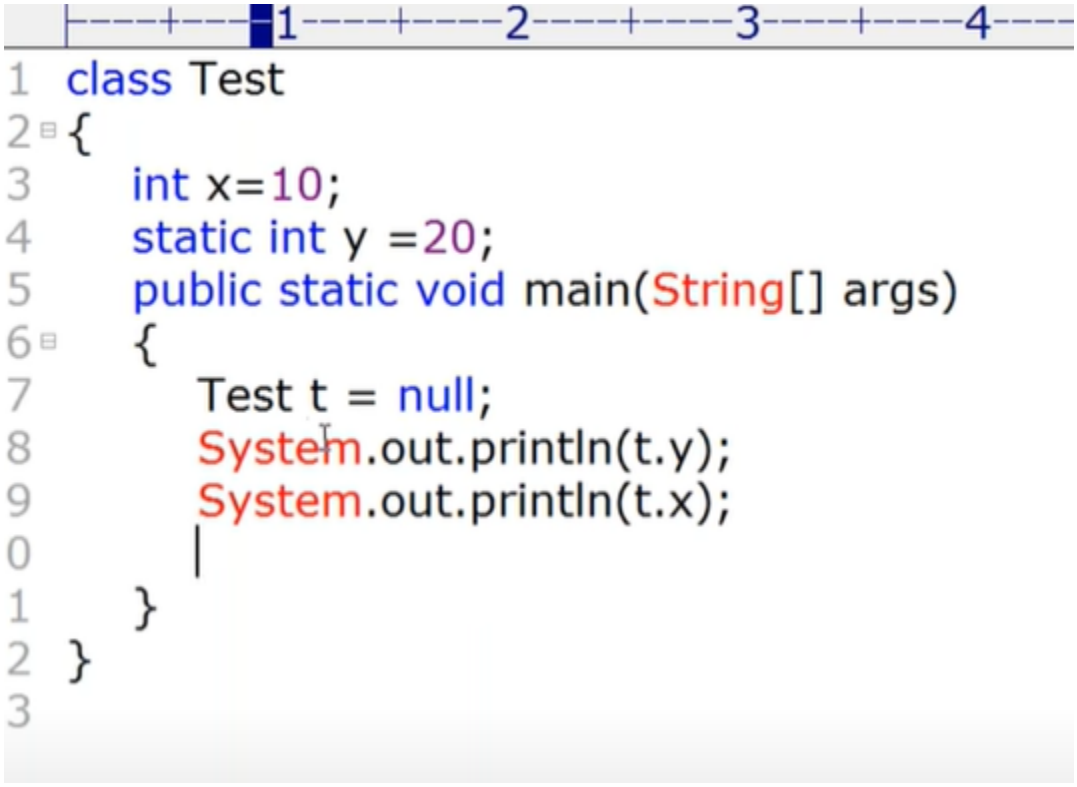
```
1  class Test
2  {
3      int x=10;
4      static int y =20;
5      public static void main(String[] args)
6      {
7          Test t1 = new Test();
8          t1.x=888;
9          t1.y=999;
0          Test t2 = new Test();
1          System.out.println(t2.x+"..."+t2.y);//10...999
2      }
3  }
4
```



```
1  class Test
2  {
3      int x=10;
4      static int y =20;
5      public static void main(String[] args)
6      {
7          Test t = null;
8          System.out.println(t.y);
9          //System.out.println(t.x);
10
11     }
12 }
13
```

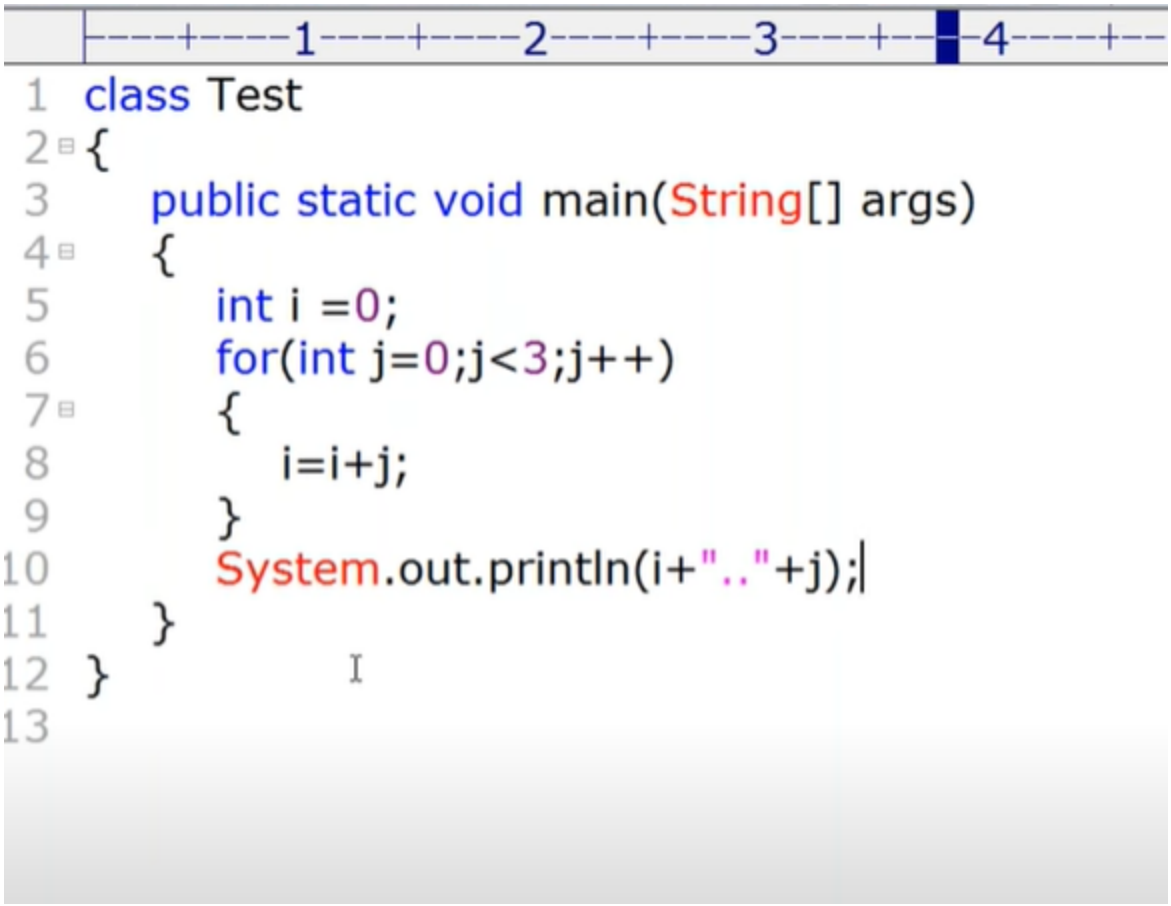
Not null pointer exception because static variable don't need any object level information

res=>20



```
1  class Test
2  {
3      int x=10;
4      static int y =20;
5      public static void main(String[] args)
6      {
7          Test t = null;
8          System.out.println(t.y);
9          System.out.println(t.x);
0
1      }
2  }
3
```

res=> 20 nullpointerException


1 class Test
2 {
3 public static void main(String[] args)
4 {
5 int i =0;
6 for(int j=0;j<3;j++)
7 {
8 i=i+j;
9 }
10 System.out.println(i+".."+j);
11 }
12 }
13

```java
class Test
{
    public static void main(String[] args)
    {
        try
        {
            int x =Integer.parseInt("ten");
        }
        catch (NumberFormatException e)
        {
            x=10;
        }
        System.out.println(x);
    }
}
```

```
D:\durgaclasses>javac Test.java
Test.java:11: error: cannot find symbol
                        x=10;
                        ^
  symbol:   variable x
  location: class Test
Test.java:13: error: cannot find symbol
                System.out.println(x);
                                   ^
  symbol:   variable x
  location: class Test
2 errors
```