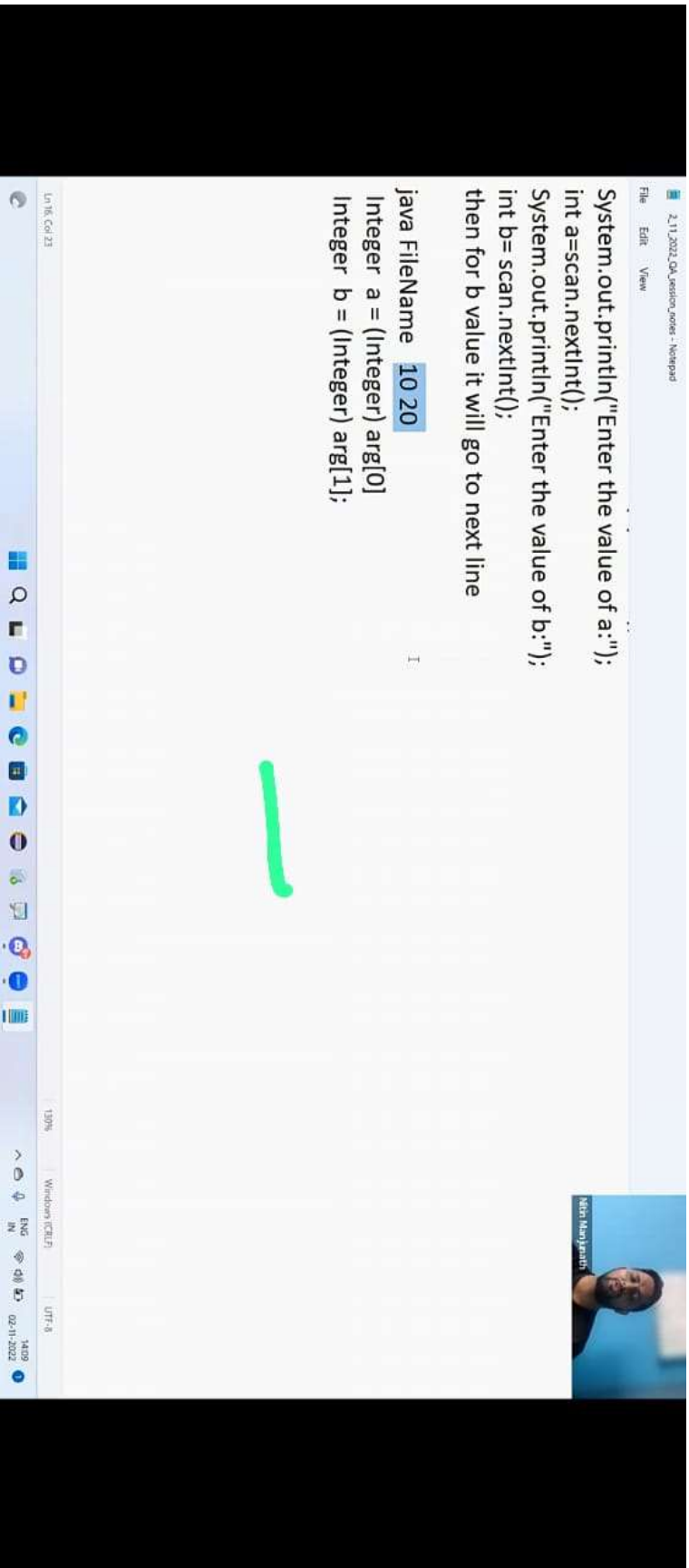


# Java String Part2





File

View

Image

Tools

Brushes

Shapes

Size

Colors

Clipboard

Image

Tools

Brushes

Shapes

Size

Colors

```
String s1=new String("dhoni");
String s2=new String("dhoni");
System.out.println(s1==s2); //false

String s3="dhoni";
String s4="dhoni";
System.out.println(s3==s4); //true
```

Stack

HeapArea (duplicates permitted)

s1

s2

s3

s4

dhoni

dhoni

dhoni

dhoni

SCP (duplicates not permitted)

memory will be cleaned at the time of jvm shutdown

144.50m

523 x 46m

340 x 340m

25°C

Brandy Cloudy

100%

ENG

IN

15:47

03-11-2022



Note:: Object creation in SCP is always optional, 1st JVM will check is any object already created with required content or not.

If it is already available then it will reuse the existing object instead of creating the new Object.

If it is not available only then new object will be created, so we say in SCP there is no chance of existing 2 objects with the same content. In SCP duplicates are not permitted.

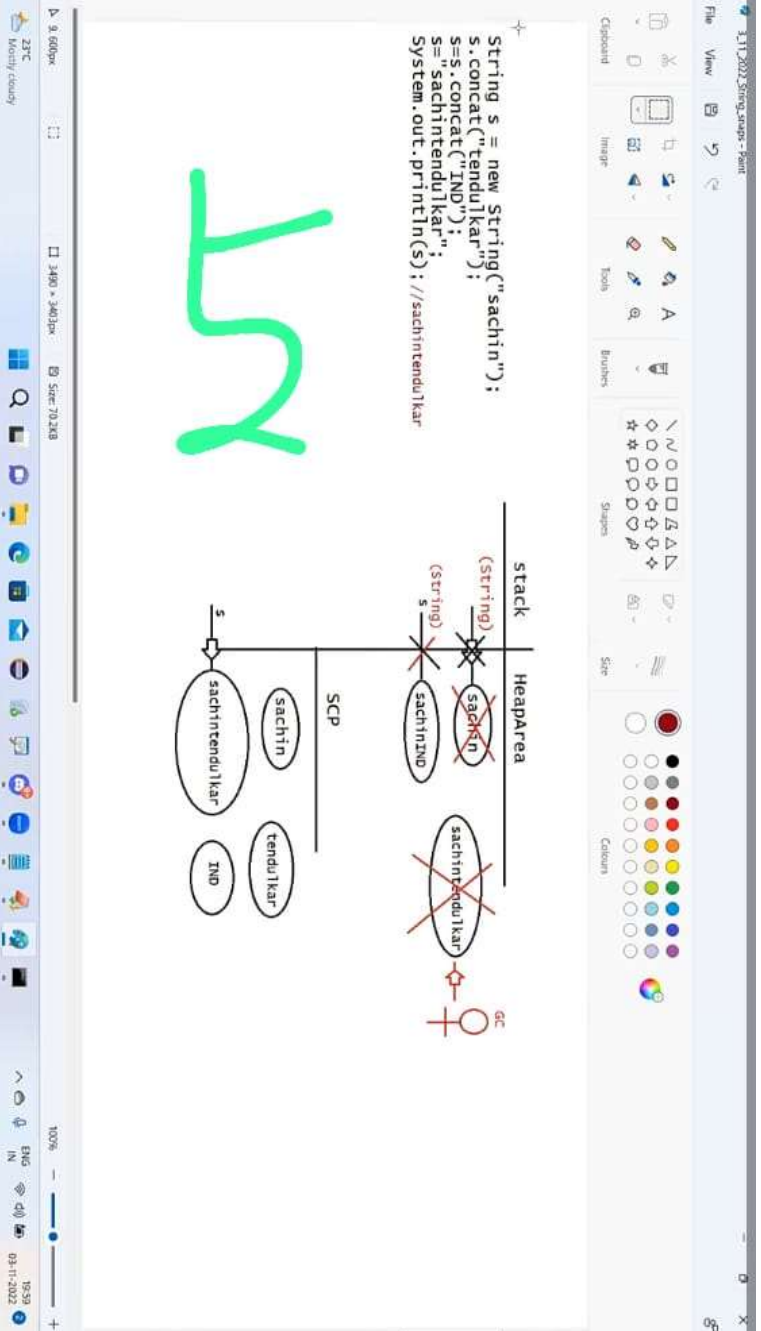
Garbage Collector cannot access SCP Area, Even though Object does not have any reference still object is not eligible for GC.  
All SCP objects will be destroyed only at the time of JVM Shutdown.

```
eg:: String s1=new String("dhoni");  
String s2=new String("dhoni");  
String s3="dhoni";  
String s4="dhoni";
```

Output:: Two objects are created in the heap with data as "dhoni" with reference as S1,S2

One object is created in SCP with the reference as S3,S4.

4



content. In SCP duplicates are not permitted.

Garbage Collector cannot access SCP Area, Even though Object does not have any reference still object is not eligible for GC.  
All SCP objects will be destroyed only at the time of JVM ShutDown.

```
eg:: String s1=new String("dhoni");  
String s2=new String("dhoni");  
String s3="dhoni";  
String s4="dhoni";
```

Output:: Two objects are created in the heap with data as "dhoni" with reference as S1,S2  
One object is created in SCP with the reference as S3,S4.

```
case 4:: String s = new String("sachin");  
s.concat("tendulkar");  
s=s.concat("IND");  
s="sachintendulkar";
```

Output:: Direct literals are always placed in SCP, Because of runtime operation if object is required to create compulsorily that object should be placed on the Heap, but not on SCP.

3.11.2022 String, StringBuffer - Print

FileView

ClipboardImageToolsBrushesShapesSizeColors

String s1= new String("sachin");  
s1.concat("tendulkar");  
s1+="IND"; s1 = s1 + "IND";  
String s2=s1.concat("MI");  
  
System.out.println(s1); //sachinIND  
System.out.println(s2); //sachinINDMI

Stack

Heaparea

s1

sachin

sachinINDtendulkar

s2

sachinINDMI

sachin

tendulkar

MI

total objects = 8(4+4)  
eligible for gc = 2

String + "IND"

addition  
both operands are number type  
addition  
if one operand is string then concatenation

408 x 66px340 x 340pxSep 20, 2018

100%ENGd/201303-11-2022

```
3_11_2022_String_session_Caracteres - Notepad
File Edit View
Q>
String s1=new String("you cannot change me!");
String s2=new String("you cannot change me!");
System.out.println(s1==s2);

String s3="you cannot change me!";
System.out.println(s1==s3);
String s4="you cannot change me!";
System.out.println(s3==s4);

String s5="you cannot " + "change me!";
System.out.println(s3==s5);

String s6="you cannot ";
String s7=s6+"change me!";
System.out.println(s3==s7);

final String s8="you cannot ";
String s9=s8+"change me!";
System.out.println(s3==s9);
System.out.println(s6==s8);

Ln 37, Col 28
23°C Mostly cloudy
Windows (CELF) UTF-8
100%
2015 03-11-2022
```



3.11.2022, String, Arrays - Print

FileView

Clipboard

Image

Tools

Brushes

Shapes

Size

Colors

Lucida Console

12

B

I

U

S

Background fill

String s1=new String("you cannot change me!");  
String s2=new String("you cannot change me!");  
System.out.println(s1==s2);//false  
  
String s3="you cannot change me!";  
System.out.println(s1==s3);//false  
String s4="you cannot change me!";  
System.out.println(s3==s4);//true  
  
String s5="you cannot " + "change me!";  
System.out.println(s3==s5);//true  
  
String s6="you cannot ";  
String s7=s6+"change me!";  
System.out.println(s3==s7);//false  
  
final String s8="you cannot ";  
String s9=s8+"change me!";  
System.out.println(s3==s9);//true  
System.out.println(s6==s8);//true

stack

heap area

s1

you cannot change me!

s2

you cannot change me!

s7

you cannot change me!

s3

you cannot change me!

s4

you cannot change me!

s9, s5

you cannot

s8

change me!

6.8.2126px

1480 x 343px

Src: 313A3

100%

2023  
03-11-2022

```
String s6="you cannot",  
String s7=s6+"change me!";  
System.out.println(s3==s7);
```

```
final String s8="you cannot",  
String s9=s8+"change me!";  
System.out.println(s3==s9);  
System.out.println(s6==s8);
```

```
output  
false  
false  
true  
true  
false  
true  
true
```

I

10

3.11.2022, Jyoti, session, classmate - Notepad

File Edit View

false  
false  
true  
true  
false  
true  
true

I

case7 :: Interning=> Using Heap object reference, if we want to get Corresponding SCP Object, then we need to use intern() method.

eg1::

```
String s1=new String("sachin");// One in heap(s1) and the other one in SCP  
String s2=s1.intern();//using s1 access object in SCP which has no reference  
System.out.println(s1==s2);//false  
String s3="sachin";  
System.out.println(s2==s3);//true
```

Ln 77, Col 1

23°C Mostly cloudy



100%

Windows (CTRL)

UTF-8

20:47 03-11-2022

3.11.2022, String, Arrays - Print

FileView

Clipboard

Image

Tools

Brushes

Shapes

Size

Colors

Lutcia Console

12B I U S Background fill

```
String s1 = new String("sachin");
String s2 = s1.concat("tendulkar");
String s3 = s2.intern();
String s4 = "sachintendulkar";
System.out.println(s3==s4);
```

12

stack

HeapArea

s1

s2

s3

s4

sachin

sachintendulkar

tendulkar

SCP

100%

20:54

03-11-2022

File Edit View

```
String s1=new String("sachin");// One in heap(s1) and the other one in SCP  
String s2=s1.concat("IND");// One in SCP(IND) and the other one in heap(s2)  
String s3=s2.intern();  
String s4="sachinIND",  
System.out.println(s3 == s4);//true
```

#### Importance of SCP

=====

1. In our program if any String object is required to use repeatedly then it is not recommended to create multiple object with same content it reduces performance of the system and effects memory utilization.
2. We can create only one copy and we can reuse the same object for every requirement. This approach improves performance and memory utilization we can achieve this by using "scp".
3. In SCP several references pointing to same object the main disadvantage in this approach is by using one reference if we are performing any change the remaining references will be impacted. To overcome this problem sun people implemented immutability concept for String objects.
4. According to this once we creates a String object we can't perform any changes in the existing object if we are trying to perform any change a new String object will be created hence immutability is the main disadvantage of scp.



FileViewPrint

3.11.2022, Sunday, 10:45 - Paint

ClipboardImageToolsBrushesShapesSizeColors

Need of SCP

Adhar Card Application

NAME

FATHERNAME

DOB

CITY

PHOTO

submit

p1

p2

p1cr

Bengaluru

(Immutable)  
(String)  
SCP AREA

p3

hyderabad

HeapArea

14

100%28/202203-11-2022

D:\>javap java.lang.String

IS

utilization we can achieve this by using "scp".

3. In SCP several references pointing to same object the main disadvantage in this approach is by using one reference if we are performing any change the remaining references will be impacted. To overcome this problem sun people implemented immutability concept for String objects.

4. According to this once we creates a String object we can't perform any changes in the existing object if we are trying to perform any change a new String object will be created hence immutability is the main disadvantage of scp.

#### String class Constructor

```
=====
String s = new String()           => Creates an Empty String Object
String s = new String(String literals) => Creates an Object with String literals on Heap
String s = new String(StringBuffer sb) => Creates an equivalent String object for StringBuffer
String s = new String(char[] ch)    => Creates an equivalent String object for character array
String s = new String(byte[] b)     => Creates an equivalent String object for byte array
```

16

```
1 public class Test {
2
3     public static void main(String[] args) {
4
5         char[] ch = {'j', 'a', 'v', 'a'};
6         String s1 = new String(ch);
7         System.out.println(s1); //java
8
9         System.out.println();
10
11         byte[] b = {65, 66, 67, 68};
12         String s2 = new String(b);
13         System.out.println(s2); // "ABCD"
14
15     }
16
17 }
18
19
20
21
22
```

5

Note: if the class name and method name is same in a class then that method is called "Constructor".

### String class Constructor

=====

String s = new String()

=> Creates an Empty String Object

String s = new String(String literals)

=> Creates an Object with String literals on Heap

eg: String str = new String("sachin");

String s = new String(StringBuffer sb)

=> Creates an equivalent String object for StringBuffer

String s = new String(char[] ch)

=> Creates an equivalent String object for character array

String s = new String(byte[] b)

=> Creates an equivalent String object for byte array

eg:

char[] ch = {'i','a','v','a'};

String s1 = new String(ch);

System.out.println(s1); //java

System.out.println();

byte[] b = {65,66,67,68};

String s2 = new String(b);

System.out.println(s2); // "ABCD"

81

eg: String str = new String("sachin");

String s = new String(StringBuffer sb) => Creates an equivalent String object for StringBuffer

String s = new String(char[] ch) => Creates an equivalent String object for character array

String s = new String(byte[] b) => Creates an equivalent String object for byte array

eg:

char[] ch = {'j','a','v','a'};

String s1 = new String(ch);

System.out.println(s1); //java

System.out.println();

byte[] b = {65,66,67,68};

String s2 = new String(b);

System.out.println(s2); // "ABCD"

StringBuffer sb = new StringBuffer("sachin");

System.out.println("StringBuffer data is :: "+sb);

String s1 = new String(sb);

System.out.println("String data is : "+s1);

61

### Important methods of String

- 1. public char charAt(int index)
- 2. public String concat(String str)
- 3. public boolean equals(Object o)
- 4. public boolean equalsIgnoreCase(String s)
- 5. public String substring(int begin)
- 6. public String substring(int begin, int end)
- 7. public int length()
- 8. public String replace(char old, char new)
- 9. public String toLowerCase()
- 10. public String toUpperCase()
- 11. public String trim()
- 12. public int indexOf(char ch)
- 13. public int lastIndexOf(char ch)

20

3.11.2022 Shiny, Kirti - Print

File View

Clipboard

Image

Tools

Brushes

Shapes

Size

Colours

Lucida Console

14

B I U S

Background fill

```
string s = new string("sachin");  
System.out.println(s[3]); //CE  
System.out.println(s.charAt(3)); //h  
System.out.println(s.charAt(-1)); //StringIndexOutOfBoundsException  
System.out.println(s.charAt(500));
```

Diagram illustrating memory layout for the string "sachin":

A pointer variable `s` points to a memory block labeled "HeapArea". The memory block contains the characters 's', 'a', 'c', 'h', 'i', 'n' stored in a contiguous array. The indices 0 through 5 are shown above the corresponding characters.

2

D: 2.4G, 89%  
C:\Program Files\Java\jdk-8.0.600\bin\java.exe

575 x 77px  
1490 x 3810px  
Size: 39.3 KB

100%

2754  
03-11-2022

Which among the following declaration is valid?

1. `int[] a;`
2. `int a[];`
3. `int []a;`
4. `int[6] a;`//can't specify the size

Predict the answer.

- A. 1,2,3
- B. 1,2,4
- C. 2,3,4
- D. None of the above

Answer: A

Q>

1. `int[] a;`  
`a=new int[5];`
2. `int[] a=new int[5];`

Both the declarations are they same?

- A. yes
- B. no

Ln 13, Col 10

Q>

1. `int[] a;`  
`a=new int[5];`
2. `int[] a=new int[5];`

Both the declarations are they same?

- A. yes  
B. no

Answer: A

23

Untitled - Paint

File View

Clipboard

Image

Tools

Brushes

Shapes

Size

Colors

Background fill

```
int n[][] = {{1,3},{2,4}};
for(int i=n.length-1;i>0;i--){
    for(int y:n[i]){
        System.out.print(y);
    }
}
```

i = 2 - 1 = 1  
1 > 0 (true)  
for(int y : n[1])  
 print(y) // 2 4

i = 0  
0 > 0 (true)  
for(int y : n[0])  
 print(y) // 1 3

n

0 1

1 3

2 4

0 1

24

202/28/2022 5:15:20 PM 1400 x 3810px

24

100%

ENG

22:19

03/11/2022

Answer: A

Q>

```
3. int n[][] = {{1,3},{2,4}};  
for(int i=n.length-1;i>=0;i--){  
    for(int y:n[i])  
        System.out.print(y);  
}
```

- A. 1234
- B. 2313
- C. 3142
- D. 4231
- E. 2413

- F. compilation error
- G. Some problem by jvm at the runtime

Answer: E

252

3.11.2022, Script, session, images - Paint

File View

Clipboard

Image

Tools

Brushes

Shapes

Size

Colors

Lutida Console

14 B I U S

Background fill

100%

22:25 03-11-2022

```
int nums1[] = {1,2,3};
int nums2[] = {1,2,3,4,5};
nums2 = nums1;
for(int x:nums2)
    System.out.print(x+" ");
```

nums1

0 1 2

1 2 3

nums2

0 1 2 3 4

0 1 2 3 4

26

```
count++;  
}  
System.out.println(count+" found");  
}
```

What is the result?

- A. Compilation fails
- B. 0 found
- C. 1 found
- D. 3 found
- E. 2 found

```
data[0]= 2010  
data[1]= 2013  
data[2]= 2014  
data[3]= 2015  
data[4]= 2014
```

```
key = 2014, count= 0  
e = 2010
```

Answer: B

2

3.11.2022\_jaypeth\_session\_classmate - Notepad

File Edit View

E. 2 found

```
data[0]= 2010
data[1]= 2013
data[2]= 2014
data[3]= 2015
data[4]= 2014
```

```
key = 2014, count= 0
e = 2010
```

Answer: B(at any time count value will not change becoz of continue, or even if condition fails)

28

Ln 82, Col 97

23°C  
Mostly cloudy



100%

Windows [CTRL]

UTF-8

22:31  
03-11-2022

3.11.2022, ScriptSession, Image - Paint

File View

Clipboard

Image

Tools

Brushes

Shapes

Size

Colors

```
int numbers[];  
numbers = new int[2];  
numbers[0] = 10;  
numbers[1] = 20;  
numbers = new int[4];  
numbers[2] = 30;  
numbers[3] = 40;  
for (int x: numbers)  
    System.out.println(" " + x);  
  
// 0 0 30 40
```

~~numbers~~

0	1	2	3
<del>10</del>	<del>20</del>		

numbers

0	1	2	3
0	0	30	40

29

25°C  
Breezy Cloudy

1440 x 3840  
Sep 10, 5:48

100%

ENG  
IN  
03-11-2022

```
class Test{  
    public static void main(String[] args){  
        int numbers[];  
        numbers = new int[2];  
        numbers[0] = 10;  
        numbers[1] = 20;  
        numbers = new int[4];  
        numbers[2] = 30;  
        numbers[3] = 40;  
        for(int x: numbers)  
            System.out.print(" " + x);  
        }  
    }
```

What is the result?

- A. 10 20 30 40
- B. 0 0 30 40
- C. Compilation fails
- D. An exception is thrown at runtime

Answer: B

00

Q>

```
int wd = 0;
String days[] = {"sun", "mon", "wed", "sat"};
for(String s:days){
    switch(s){
        case "sat":
            wd-=1;
        case "sun":
            wd-=1;
            break;
        case "mon":
            wd++;
        case "wed":
            wd+=2;
    }
}
```

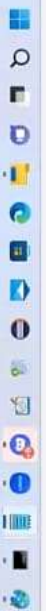
What is the result?

- A. 3
- B. 4
- C. -1
- D. compilation fails

JVM

Ln 134, Col 32

23°C Mostly cloudy



100%

Windows (CTRL)

UTF-8

22:43 03-11-2022

3.11.2022\_javappt\_session\_4classmate - Notepad  
File Edit View

```
wd+=2;
```

```
}
```

What is the result?

- A. 3
- B. 4
- C. -1
- D. compilation fails

JVM

===

```
days[0] = "sun", days[1] = "mon", days[2] = "wed", days[3] = "sat"
```

```
wd = 0, s = "sun", change wd = wd-1 = 0-1 = -1
```

```
wd = -1, s = "mon", wd = -1+1 = 0, wd = 0+2 = 2
```

```
wd = 2, s = "wed", wd = 4
```

```
wd = 4, s = "sat", wd = 4-1 = 3
```

Answer: A

52

=> concat(String)  
=> String object by default Immutable(changes made will reflect in new memory)

```
String[] str = {"A", "B"};  
int idx = 0;  
for(String s: str){  
    str[idx].concat(" element " + idx);  
    idx++;  
}  
//using for loop  
for(idx=0;idx<str.length;idx++){  
    System.out.print(str[idx]);  
}
```

What is the result?

- A. AB
- B. A element0B element1
- C. A NullPointerException is thrown at runtime
- D. A 0B 1
- E. Compilation Error

Answer: A

53

FileViewImageToolsBrushesShapesSizeColors

```
int[][] arr = new int[2][4];
arr[0] = new int[]{1, 3, 5, 7};
arr[1] = new int[]{1, 3};
for (int[] a : arr) {
    for (int i : a) {
        System.out.print(i + " ");
    }
    System.out.println();
}
```

[20]arr

01

0000

0000

13

[10]

34

23°C

42 × 49px

440 × 300px

Size: 100%

ENG

22:56

03-11-2022

