

~~extra credit~~ ~~class~~ ~~object~~ ~~while~~
~~arithmetic~~ ~~not~~ ~~for~~ ~~statement~~
~~operator~~ ~~statement~~ ~~statement~~
~~for~~ ~~statement~~

Range

$$\text{byte} = [-2^7] \text{ to } [+2^7 - 1]$$

$$\text{short} = [-2^{15}] \text{ to } [+2^{15} - 1]$$

$$\text{int} = [-2^{31}] \text{ to } [+2^{31} - 1]$$

4-5-23

lect 18: for loop

Page No.: /

Date: / /

5-4-23

class forDemo {

public static void main(String[] args) {

int x = 10;

for (int i = 1; i <= 10; i++) {

sop(x + i);

}

}

O/P → 1 2 3

for loop
for
for
for

int i
int i
int i
int i

compile
time
error
syntax
error

error: cannot find symbol

sop(x + i);

symbol: variable

location: class forDemo.

→ array किए अलग हैं for loop का use करें।

int x = 10;

int i = 1;

for (; i <= 10; i++) {

{

sop(x + i);

}

O/P ⇒ 1 2 3

1 to 1000 find
perfect number
Total 31 nos.
31st question 81st min.

- ① Palindrome number
② prime number
③ perfect numbers

Page No.: / / Date: / /

Q.5. Take N as input. Count all its factors and print count

Q.6. Take N as input. Print whether it's prime or not.

Quiz: If number (from 1 to N) is divisible by 1 and itself is called Prime Number.

A. True

$$1 \nmid 1 \dots = 0$$

B. false. ✓

$$1 \nmid 1$$

because 1 नहीं prime number nahi
but 1 का defⁿ अस्ति 1 एवं 1 एवं
prime number नहीं.

→ actual defⁿ :- prime number अस्ति number
अस्ति जिसके फैक्टर्स exactly दो
अस्ति

Code of prime number

```
int N = 5
```

```
int count = 0;
```

```
for(int i=1; i<=N; i++) {
```

```
    if (N % i == 0) {
```

```
        count++;
```

```
}
```

```
if (count == 2) {
```

```
    Sop("prime")  
} else {
```

```
Sop("not prime");
```

$$h \xrightarrow{x} \frac{1+2}{3} h$$

$$\textcircled{6} \rightarrow \frac{1+2+3}{6}$$

Q. 1. Take integer N as input. Print whether N is perfect number or not.

$$i/p : 4$$

$6|P$: not a perfect number

i/p : 6

Perfect number

$$\int N = 4$$

~~int count =~~

int sum=0; step approach

```
for (int i=1; i<=N; i++){
```

if ($N \neq 0$, $i = 0$) {

sum = sum + i;

if ($\text{sum} \neq N$) {

SOP(~~it is~~ "perfect number");

} else {

$\text{S}\ddot{\text{o}}\text{p}(\text{"not a perfect number"})$;

$$\underline{N=6}$$

$$i=1 \quad i < 6 \quad N \neq 0 \quad \text{sum} = \quad i^{\text{th}}$$

~~\$1 C6 6% = 0.6 sum = 1 2~~

$$2 < 6 \quad 6 \% 2 == 0 \vee \text{sum} = 3$$

$$3 \times 6 = 18 \quad \text{sum} = 6$$

$$4 \times 6 \text{ log } y = 20x \quad \text{sum} = 6 \quad 5$$

$$5 < 6 \quad 6 \text{ or } 5 = 20x \quad \text{sum} = 6$$

$$\frac{6 \times 6}{\cancel{6} \times \cancel{6}} = \frac{36}{1}$$

~~ST~~ ~~any~~
run
Siruni
sanctimony
approach ||
31/8.

Q.8. Take N as input. Print whether N is Armstrong number or not.

Input: 23

O/p : - Not an Armstrong Number

Typ: 153

Opp: Armstrong Number.

Code:

int N = 153
int temp = N;
int sum = 0;

for (int i=1; i<=N; i++) {

 int rem = N%10;

 sum = sum + rem*rem*rem;

 N = N/10;

}

if (sum == temp) {

 SOP("it is")
 SOP("Armstrong Number");

? else {

 SOP("Not an Armstrong number");

N = 153;

int sum = 0;

int temp = N;

while (N != 0) {

 int rem = N % 10;

 sum = sum + rem*rem*rem;

 N = N/10;

}

sum = 153

5-4-23

for loop 03

quiz explanation

for loop ~~not~~ ~~at~~ condition ~~unify~~
~~at 24/7/19~~

Armstrong number \Rightarrow

$$13 \Rightarrow 2\text{digit} \Rightarrow 1^2 + 3^2 = 13 \times$$

$$153 \Rightarrow 3\text{digit} \Rightarrow 1^2 + 5^3 + 3^3 = 153 \checkmark$$

$$1634 \Rightarrow 4\text{digit} \Rightarrow 1^4 + 6^4 + 3^4 + 4^4 =$$

```
int N = 153 or N
int temp1 = N 3; temp2 = 153 or N
int count = 0
```

(1) $\text{while}(N \neq 0)$ {

~~int~~ ~~temp~~ count ++;

~~N = N / 10;~~

}

~~int sum = 0.~~

$\text{white}(\text{temp}) \neq 0$ {

int rem = ~~temp % 10;~~

~~sum = sum + rem~~

int mult = 1;

for (int i=1; i < count; i++) {

~~int rem = temp % 10;~~

~~int mult = 1;~~

~~for (int i=1; i < count; i++) {~~

sum = sum + mult

~~N = N / 10;~~

}

Date: / /

```
if (temp2 == sum) {
    sop("Armstrong");
} else {
    sop("Not a Armstrong");
}
```

(3)

* Scope of Variables => *

{

~~function
scope
block~~
~~same name~~
~~variable name~~

```
int x = 10;
```

```
int x = 20
```

}

~~function
scope
block~~
~~same name~~
~~variable name~~

{

```
int x = 10;
```

```
x = 20;
```

}

Q. 2. Predict the o/p

```
class Core2Web {
```

[]

```
    public static void main (String [] args) {
```

```
        int x = 10, y = 20;
```

{

```
        sop (x + " " + y);
```

10 20

{

```
x = 15;
```

```
sop (x + " " + y);
```

15 20

{

```
sop (x + " " + y);
```

15 20

{

?

Q.3

class Core2Web

public static void main(String[] args) {

int x = 10;

x [10]

{

int y = 20;

y [20]

SOP(x + " " + y); // 10 20

{

SOP(x + " " + y);

{

SOP(x + " " + y);

}

↑

ans: D ⇒ 2 error

a. class Core2Web {

public static void main(String[] args) {

int x = 10;

x [10]

{

int y = 20;

y [20]

{

SOP(x + " " + y); // 10 20

{

int y = 15; y [15]

{

SOP(x + " " + y); // 10 15

}

SOP(x + " " + y);

↑ error

O/P → 1 error

c. class (overwriting)

public static void main (String [] args) {

int x = 10; x 10

}

int x = 20; x 20

System.out (x)

2 option

}

{

int x = 30; x 30 2 option

System.out (x);

static void main (

Preference

} 3

{

System.out (x);

1 option.

3

o/p :
20
20, 30
10

lect 32

26-5-23 seen
26-5-23 uprooted
Date: 26-5-23
Data: 26-5-23
Java Date 26-5-23

class Dcm {

public static void main (String [] args) {

```
    Demo obj = new Demo();
    obj.fun(10);
    obj.fun(10.5f);
```

}

Void fun (int x) {

```
    System.out.println("In fun");
    System.out.println(x);
```

}

In fun

obj : 10

In fun

10

compile time error: incompatible types:
possible lossy conversion from float
to int

obj.fun(10.5f);

Note: Some messages have been simplified;
recompile with -Xdiags:verbose to get
full output

class Demo

public static void main(String[] args) {

 Demo obj = new Demo();

 obj.fun(10);

 obj.fun(10.5f);

 obj.fun(true);

}

 void fun(float x) {

 System.out.println("In fun");

 System.out.println(x);

}

O/P: In fun

10.0

In fun

10.5

In fun

65.0

 obj.fun(true);

error: incompatible types: boolean cannot be converted to float.

* Return Value ↗

class Demo {

int fun (int x) {

return x+10;

}

public static void main (String args) {

Output ↗

20

Demo obj = new Demo();

System.out.println (obj.fun(10));

class Demo {

void ~~int~~ fun (int x) {

int y = x+10;

}

public static void main (String args) {

Demo obj = new Demo();

System.out.println (obj.fun(10));

}

* Error: void type not allowed here

System.out.println (obj.fun(10));

Q. class Demo {

void fun (int x) {

int y = x + 10;

}
public static void main (String [] args) {

Demo obj; obj.fun(10);

int a = obj.fun(10);
System.out.println(a);

* Error's incompatible type of void
cannot be converted to int

int a = obj.fun(-10);

VII
Sheet 33

seen at 27-5-23
uploaded -> 25-5-23
Date

Array of
String Data Structure.

```
class Demo1  
    int a=10;
```

```
public static void main( String args)
```

int b=20; → local variable

System.out.println(b);

System.out.println(a); → error;

}

In C

```
void main()
```

```
// int arr[]; error → size is not given.
```

}

In Java

if declaration
gets memory
it turns to be
definition (in Java)
objects (in C)

```
// int arr[]; There is no error
```

↳ it is declaration

```
arr[0]; in Java
```

↳ it gives error:

→ ~~at java start~~ array heap section
~~at start~~ start.

start heap ~~at~~ start

new ~~at~~ start.

→ ~~at~~ start heap new ~~at~~ start ~~at~~ start

→ array → dynamic memory allocation function

new ~~for dynamic~~

class ArrayDemo

```
public static void main(String[] args) {
```

```
int arr1[] = new int[7];
```

↑ // error: Array

int arr2[] = new int[4]; dimension missing

(program

```
void main() {
```

```
int arr1;
```

error

↑ size missing

- 2 way to initializing declare array

```
int arr2[] = new int[7] {10, 20, 30};
```

```
int arr2[] = {10, 20, 30, 40};
```

→ set internally same init

array for memory

array runtime allocation

- check what by default store in array into float, char object

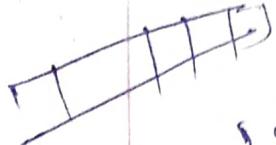
↳ that means just like upside mention array

for int, byte, short, long = 0

float, double = 0.0

char = " " blank char.

boolean = false



array - c Array Demo Class

[Main.java / Demo.java]

Lect 34 seen at 27-5-23
uploaded 26-5-23

① int arr[4] = new int[4];

arr[0] = 10;

arr[1] = 10;

arr[2] = 10;

arr[3] = 10;

② int arr2[] = {20, 20, 20};

③ int arr3[] = new int[] {30, 30, 30};

error: array
Creation with both
dimension expression

and initializ"
is illegal

// int arr4[] = new int[4] {40, 40, 40, 40};

← → error: [Can't find constant value]
size formula
list goes here.

public static void main (String[] args) {

int arr1[] = {10, 20, 30, 40, 50};

char arr2[] = {'A', 'B', 'C'};

float arr3[] = {10.5f, 20.5f};

boolean arr4[] = {true, false, true};

SOP (arr1[0]);

SOP (arr1[1]);

SOP (arr1[2]);

SOP (arr1[3]);

SOP (arr1[4]);

SOP (arr2[0]);

SOP (arr2[1]);

← SOP (arr2[2]);

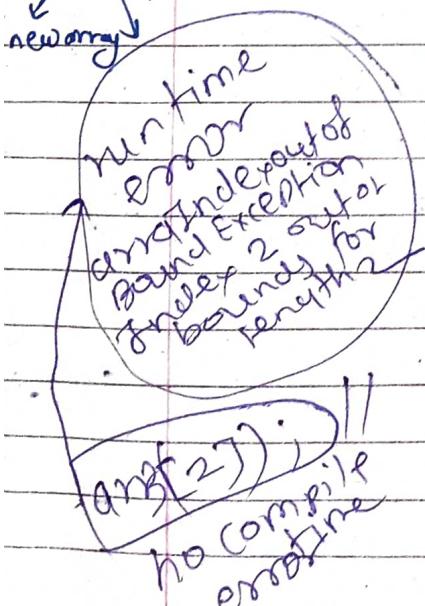
SOP (arr3[0]);

SOP (arr3[1]);

SOP (arr4[0]);

SOP (arr4[1]);

SOP (arr4[2]);



* accessing elements from an array *

class ArrayDemo {

 public static void main(String[] args)

 int arr[] = {10, 20, 30, 40, 50};

 System.out.println(arr[0]);

 System.out.println(arr[1]);

 System.out.println(arr[2]);

 System.out.println(arr[3]);

 System.out.println(arr[4]);

 System.out.println("using for loop");

 for (int i = 0; i < 5; i++) {

 System.out.println(arr[i]);

}

* class ArDemo

 int arr[] = new int[5];

 BufferedReader br = new BufferedReader

(new InputStreamReader(System.

~~arr~~

~~int D = br.~~

~~int D = Integer.parseInt(br.read~~

~~Q3~~

```
for (int i=0; i<5; i++) {
```

```
    arr[i] = Integer.parseInt(br.readLine());
```

{

```
for (int i=0; i<5; i++) {
```

```
sop(arr[i]);
```

}

Or using scanner:

```
Scanner sc = new Scanner(System.in);
```

```
for (int i=0; i<5; i++) {
```

```
    arr[i] = sc.nextInt();
```

}

```
for (int i=0; i<5; i++) {
```

```
sop(arr[i]);
```

}

- Try with different typed user input except what that given.

29-5-23
lect 35: Array⁰³

Page No.: / /
Date: / /

Q. Sum of all array elements:

```
int arr[5] = new int arr[5];  
BufferedReader br = new BufferedReader(new InputStreamReader(  
System.in));  
for (int i = 0; i < 5; i++) {  
    arr[i] = Integer.parseInt(br.readLine());  
    int sum = sum + arr[i];  
}  
System.out.println(sum);
```

Q. Printing Even and odd count of elements in an Array

```
class ArrayDemo{  
    public static void main (String[] args){  
        Scanner sc = new Scanner (System.in);  
        int size = sc.nextInt();  
        System.out.print ("enter Array size");  
        int arr[] = new arr [size];  
        int count = 0;  
        System.out.print ("enter an element");  
        for (int i = 0; i < arr.length; i++) {  
            arr [i] = sc.nextInt();  
            if (arr[i] % 2 == 0) {  
                count++;  
            }  
        }  
        System.out.print ("Even  
        System.out.println (count);  
    }  
}
```

```
sop("Even count"+count);
```

```
class ArrayDemo{
```

```
public static void main(String [] args){
```

```
Scanner sc = new Scanner(System.in)
```

```
sop("Enter size of array");
```

```
int size = sc.nextInt();
```

```
int arr[] = new int [size];
```

```
sop("Enter array elements");
```

```
for (int i=0; i<arr.length; i++) {
```

```
arr[i] = sc.nextInt();
```

```
}
```

```
int count1 = 0;
```

```
int count2 = 0;
```

```
for (int i=0; i<arr.length; i++) {
```

```
if (arr[i] % 2 == 0) {
```

```
    count1++;
```

```
else {
```

```
    count2++;
```

```
} }
```

```
sop("Even count "+ count1);
```

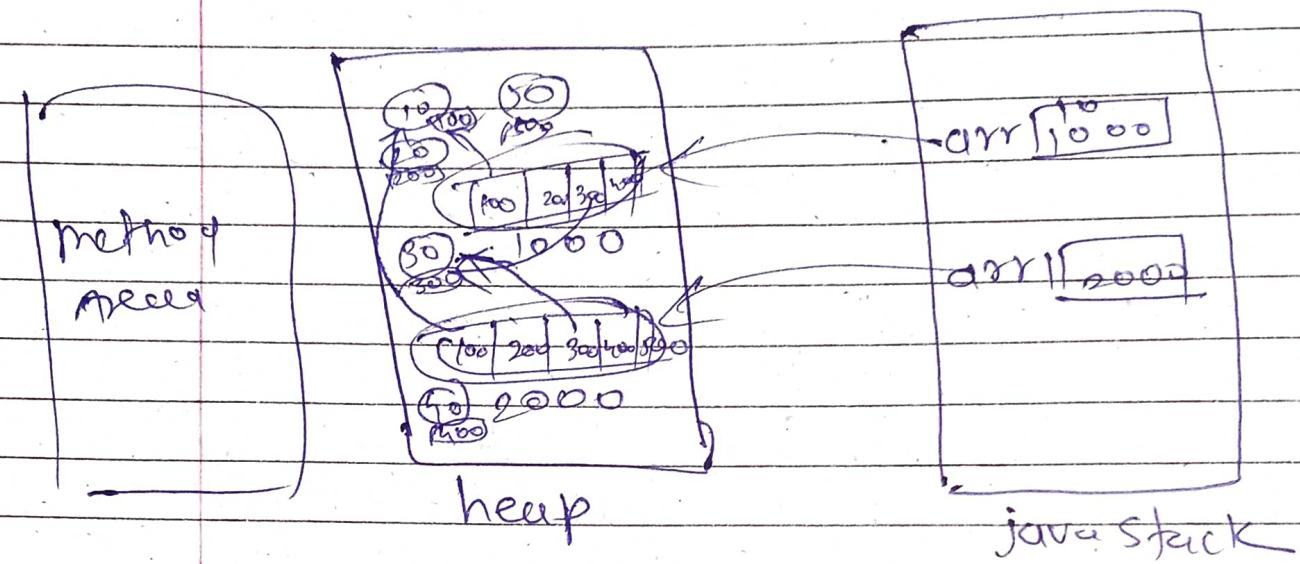
```
sop("Odd count "+ count2);
```

VM 3109 pointer
Data: 3109 3109
C++ 3109 3109

Internals of array

int arr[] = {10, 20, 30, 40, 50}

int arr[] = {10, 20, 30, 40, 50}



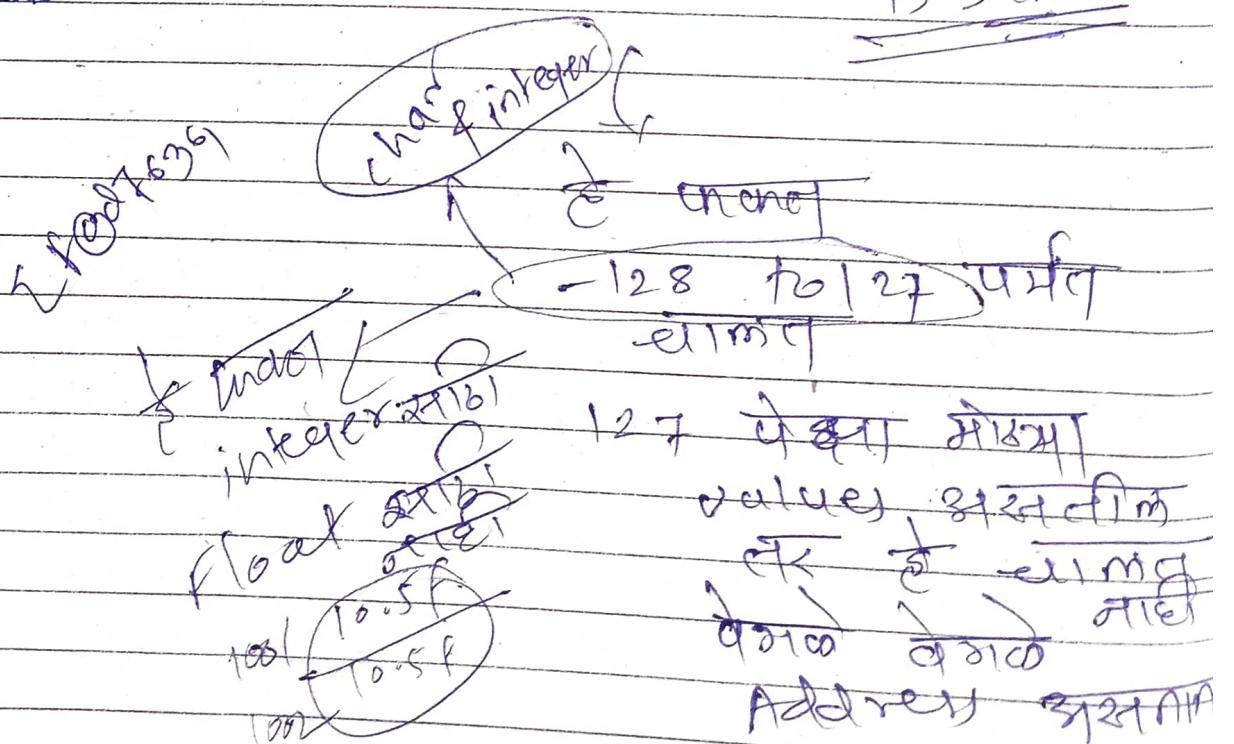
sop(arr);

boolean
and
char arr
same values

char identityHashCode
same stat.

sop(arr);

identityHashCode(arr[0]) or [arr[0]]
is same



System.identityHashCode(arr)

String constant pool

Page No.

Date

Array of

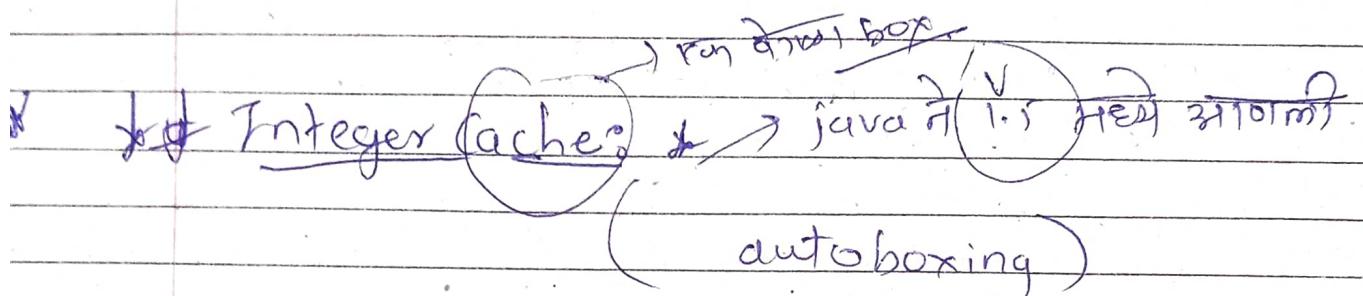
leet 36 . 30 | 05 | 23

float arrs[] = { 10.5, 20.5, 30.5 };

sop(arrs);

O/P \Rightarrow error:- possible lossy conversion.

what is
StringConstantPool



Integral

Character

Integer

see
wrapper class
source code

wrapper
class
source code

class IntegerCache {

public static void main(String args[]) {

Same
Identity
HashCode
because
autoboxing
class

{
int x = 10;
int y = 10;
Integer z = 10;

not same
for short,
byte,
char

Integer a = new Integer(10);

sop(System.identityHashCode(x));

sop((y));

sop((z));

or for int
and tydya wrapper
class salhi same asto.

* for each loop *

```
int arr[] = {10, 20, 30, 40, 50};
```

```
for (int i=0; i<arr.length; i++) {  
    sop(arr[i]);  
}
```

```
for (int x : arr) {  
    sop(x);  
}
```

```
class forEachDemo {
```

```
    public static void sum(String[] args) {
```

```
        int arr[] = {10, 20, 30, 40, 50};
```

```
        for (float x : arr) {
```

```
            sop(arr[0]);
```

```
            sop(x);  
        }
```

char arr[3] = { 'A', 'B', 'C' };
SOP (arr);
O/P: ABC → String internally another array [318m]

* Passing array as an argument *

class funDemo {

public static void main (String [] args) {

funDemo obj = new funDemo();

obj.fun();

int ret = obj.fun(10);

Prototype

}

function header

void fun() {

System.out.println("In fun");

it's get
priority
while
printing
error

O/P:

int fun(int x) {

int val = x * 50;
return val;

}

}

O/P: - 600

01-5-23 uploaded

Page No.: / /
Date: / /

* Lec 37 :- Command line argument \Rightarrow

seen 01-6-23

* C array return char in str.

Q. class Demo {

 static void fun (int x, float y)

 sop(x);
 sop(y);

 public static void main (String [] args) {

 fun (10, 20.5f);

}

Q. static void fun (int x, double y) {

 sop(x);
 sop(y);

}

 public static void main (String [] args)

 fun (10, 20);

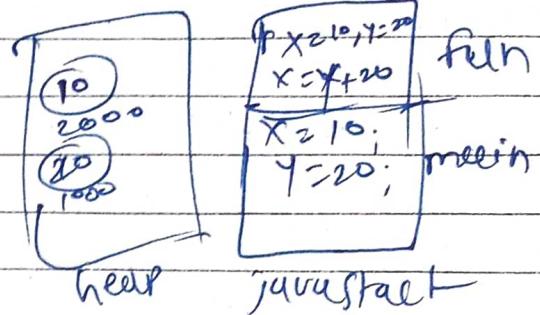
IMP 8

O/P : 65
20.0.

in \rightarrow call by value
inc/cpp \rightarrow call by reference & value also

static void fun(int x, int y) {

sop(x);
sop(y);
x = x + 10;
y = y + 10;
sop(x);
sop(y);



public static void main(String [] args) {

int x = 10;
int y = 20;
sop(x);
sop(y);
fun(x,y);
sop(x);
sop(y);

O/P :
10
20
10
20
20
30
20

static void fun(int xarr[]) {

for (int x : xarr) {

sop(x);

}

} public static void main(String [] args) {

int arr[] = { 10, 20, 30 };

for (int x : arr) {

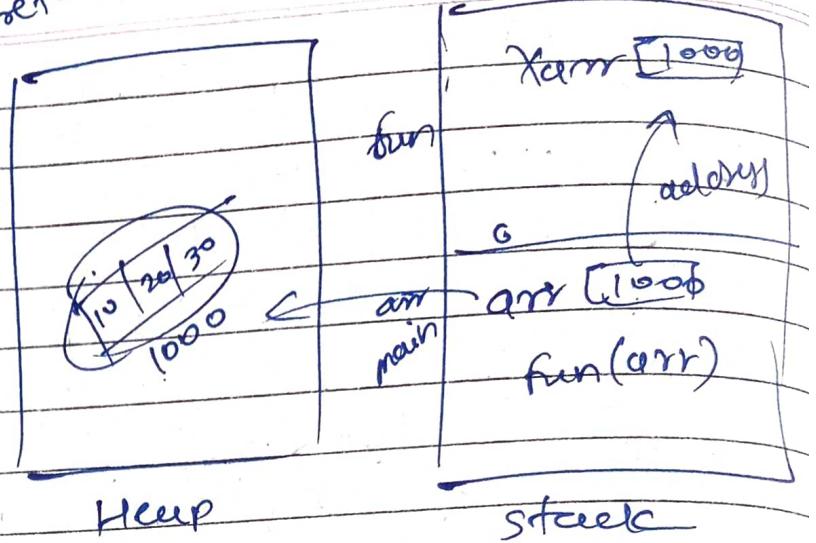
sop(x);

}

class और address
अपनी परिवर्तनी
reference वाली

Page No.: / /
Date: / /

fun(arr);

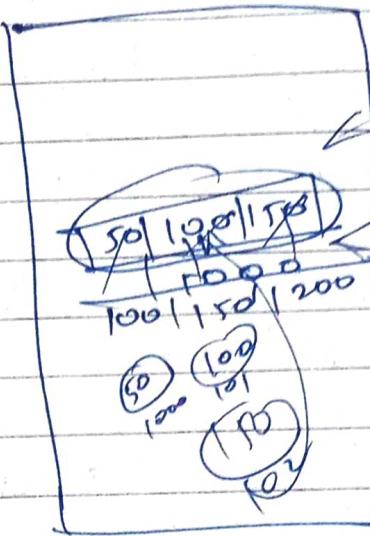


Array function ~~not~~ address is pass
by ref. → function call ~~optimized~~

```
class ArrayDemo {  
    static void fun (int arr[]) {  
        for(int x:arr) {  
            System.out.println(x);  
        }  
        for (int i=0; i<arr.length; i++) {  
            arr[i] = arr[i]+ 50;  
        }  
    }  
    public static void main (String [] args) {  
        int arr[] = { 50, 100, 150};  
        fun (arr);  
        for (int x: arr) {  
            System.out.println(x);  
        }  
    }  
}
```

O/P:

50
100
150
100
150
200



heap

~~50, 100, 150~~

fun

main

arr[1000]

int arr[1000]

fun(1000)

main
Stack.java

Command line arguments

```
class Demo {  
    public static void main (String [] args) {
```

```
        String friends [] = {"Ashish", "Kanha", "Badhe",  
                            "Rahul"};
```

```
sop (args[0]);  
sop (args[1]);  
sop (args[2]);
```

}
}

At ~~the~~ ~~at~~ run time ~~in~~ command
Host data take ~~to~~ ~~in~~ new code
run ~~in~~.

Example: javac program1.java

java Demo Kanha, Badhe, Shashi

```

class Demo {
    public static void main(String[] args) {
        String friends[] = { "Ashish", "Kanha",
                             "Bachhe", "Rahul" };
        for (int i=0; i<args.length; i++) {
            System.out.println(args[i]);
        }
    }
}

public static void main(String... args) {
    // variable no. of argument
    // or
    // ellipse
}

```

Fr 6.23
seen → 2.6.23
lect 38

python int 401
bytecode 0011

JVM on array

Hexadecimal
number

[C A F E B A B E]

magic number of java

JVM architecture.

class loader

method area

heap

java stack

PC register

native method stack

```
* class ArrayDemo{  
    void fun (int [ ] arr){  
        arr[1]=70;  
        arr[2]=80;  
    }  
    }  
    public static void main (String [ ] args){  
        int arr [ ] = { 10, 20, 30, 40 };  
        System.out.println (System.identityHashCode (arr[0]));  
        System.out.println (System.identityHashCode (arr[1]));  
        System.out.println (System.identityHashCode (arr[2]));  
        System.out.println (System.identityHashCode (arr[3]));  
  
        ArrayDemo Obj = new ArrayDemo();  
        Obj.fun (arr);  
        for (int x : arr) {  
            System.out.println (x);  
        }  
    }
```

जैव एवं मुल्टी पाइट दोनों तरीके double pointer
जैव एवं मुल्टी पाइट के लिए double pointer
like as array

int x = 70;

int y = 80;

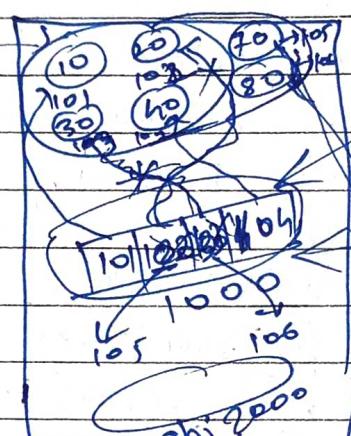
SOP(System.identityHashCode(x));

SOP(System.identityHashCode(y));

?
? ?

O/P in form of diagram

Integer
cache



heap

arr[1000]

arr[1] = 70;
arr[2] = 80;

fun()

arr[1000]

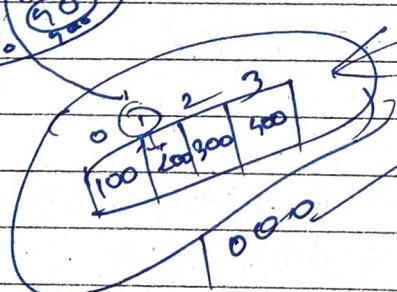
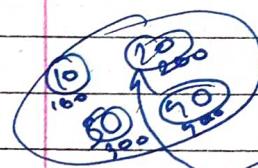
arr[1] = obj
arr[2] = fun
int x = 105
int y = 106

main()

java stack

Double Pointer Concept.

Integer
cache



arr[1000]

arr[1] → access 1000
पहला असेस 1000
address दूसरी

जोड़े 1000 address
एक असेस फ्रेमेट

index 1 जोड़े 200
दूसरा असेस 200
मेंगो. प्राप्त

concept of
double pointer

जैव।

→ java चेति फिरूज नंबर वर्गी ओपरेटर अन्य क्षेत्रों में भी उपयोग किया जाता है।
→ Collections → important in interviews of companies

new arraylist byte code check

ArrayList

ArrayList

5%

प्रश्नों में
मिलते हैं

→ सैर एवं एक्सेस कॉन्सेप्ट

→ Garbage Collector हा उच्ची लेवल डिप्लोमा के लिए बहुत ज्ञानी है।

→ heap एवं global memory बीटे।

→ सैर एवं integer cache इसकी लक्षणीय
string constant pool क्षमता।

→ Command Line Interface एवं dynamic
array तो प्रदर्शन String ने किया।

→ आपको dynamic array बहुत ज्ञानी होना चाहिए।

→ सैर command line एवं बीटे।

2-6-23

3-6-23

lect 39: 2 D array

"%P" arr
Page No.:
Date: 1/1/23

address of arr

identify
new code
↓
data arr.
संग्रह

multidimensional Array

2-D array

↑
3-D Array

`int arr [•][•] = new int [2][3];`

0	1	2
0	2 4 6	
1	2 4 16	

`arr[0][0] = 2;`

`arr[0][1] = 4;`

`arr[0][2] = 6;`

`arr[1][0] = 12;`

`arr[1][1] = 14;`

`arr[1][2] = 16;`

`int arr [] [] = new int [2][3];`

`arr[0][0] = 10; arr[1][0] = 40;`

`arr[0][1] = 20; arr[1][1] = 50;`

`arr[0][2] = 30; arr[1][2] = 60;`

0	1	2
10	20	30
40	50	60

`for (int i=0; i<2; i++) {`

`for (int j=0; j<3; j++) {`

`System.out.println (arr[i][j]);`

}

* class ~~path~~.TwoDArray{

 public static void main (String [] args){

 int arr [] [] = new int [2][2],

 arr2 [] = int [2];

 System.out.println (arr.length);

 System.out.println (arr2.length);

Page No. / /
Date: / /

multidimensional
array here
no. of rows 3
cols 2 length 6
812107

off 2 2 2

int arr[][] = new int[3][2];

→ 2 dim
variables

→ arr[][] = new int[3][3];

→ 3 dim

→ arr[][][] = new int[3][3][3];

class TwoArray {

public static void main (String [] args) {

int arr[][] = new int[2][2];

arr[0][0] = 10;

arr[0][1] = 10;

arr[1][0] = 10;

arr[1][1] = 10;

sop(arr[1][1]);

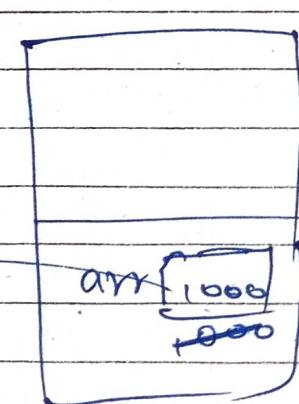
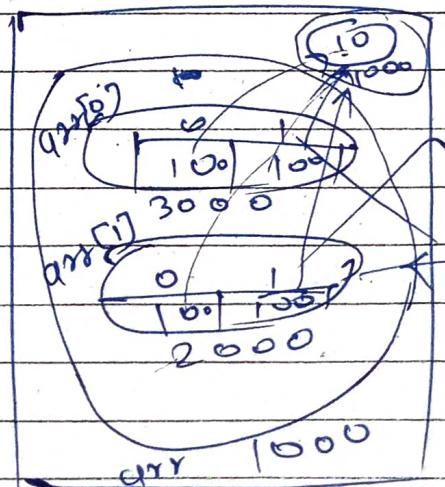
sop(arr[0]);

sop(arr[1]);

sop(arr);

new internal
memory
allocation
call current
frame.

java here 2D or 2D array internally
internally 1D array use current.



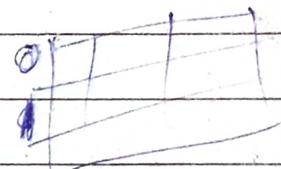
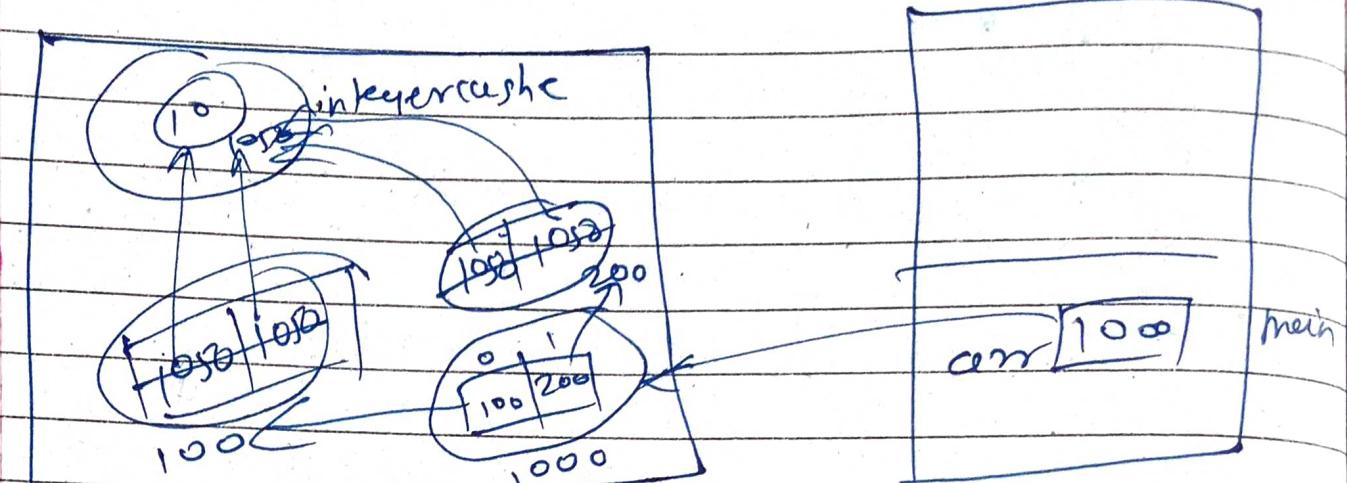
new loc
+
memory
allocation
stack

heap

see the byte code of this code backside

Page No.: / backside

Date: / /



* Tagged array

```
public static void main(String [] args) {
```

```
    int arr [][] = {{10, 20, 30}, {30, 40}, {50}}
```

```
    for (int i = 0; i < arr.length; i++) {
```

```
        for (int j = 0; j < arr[i].length; j++)
```

```
            System.out.print(arr[i][j] + " ");
```

}

```
    System.out.println();
```

O/P:

10	20	30
30	40	
50		

(rigid)

upload - 2-6-23
seen - 5-6-23

Page No.:
Date: / /

rect go

jagged array

↳ It ~~has~~ ~~memory~~ memory address.

class ArrayDemo {

public static void main(String [] args) {

int arr2[][] = {{1, 2, 3}, {4, 5}, {7, 8}};

i < arr2.length: for (int i = 0; i < 3; i++) {

j < arr2[i].length: for (int j = 0; j < 3; j++) {
sop(arr2[i][j] + " ");

?
sop();
? ? ?

O/P: 1 2 3

0	1	2
1	2	3
2		X

arr2[1][2]

error Exception ArrayIndexOutOfBoundsException

class or obj.

using for each loop.

for (int x : arr2) {

for (int y : x) {

sop(y + " ");

? ? sop(1);

Page No.
Date
.....

int arr2[][] = {{1, 2, 3}, {4, 5, 6}};
= {{new int[]{1, 2, 3}},
{new int[]{4, 5, 6}}};

jagged array initialization. → ②

class JaggedArray {

 public static void main (String [] args)

 ① int arr2[][] = {{1, 2, 3}, {4, 5, 6}};

 ② int arr[][] = new int[3][];

 arr[0] = new int[] {1, 2, 3};

 arr[1] = new int[] {4, 5};

 arr[2] = new int[] {6};

 import java.util.*;

 public static void main (String [] args) {

 int arr[][] = new int[3][];

 arr[0] = new int[3];

 arr[1] = new int[2];

 arr[2] = new int[1];

 Scanner sc = new Scanner (System.in)

 arr[0] = sc.nextInt();

 arr[1] = sc.nextInt();

 arr[2] = sc.nextInt();

 for (int r : arr)

 for (int i = 0; i < arr.length; i++)

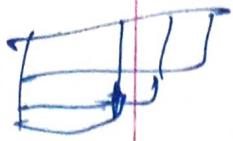
 for (int j = 0; j < arr[i].length; j++) {

 arr[i][j] = sc.nextInt();

 System.out.println (arr[i][j]);

 }

 System.out.println ();



3D Array

$$\text{Plane No.: } \begin{array}{|c|c|} \hline 0 & 0 \\ \hline \end{array}$$

$$\text{Dim: } \begin{array}{|c|c|} \hline 0 & 1 \\ \hline \end{array}$$

$$(0+0)*0 = 0$$

$$(0+1)*1 = 1$$

$$(1+0)*1 = 1$$

$$(1+1)*1 = 2$$

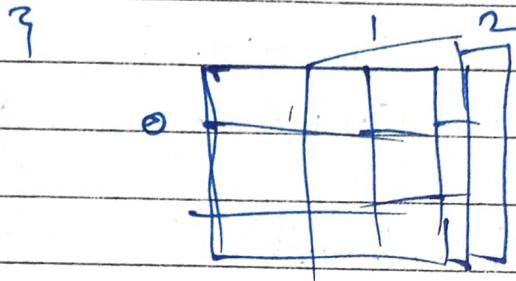
$$(2+0)*2 = 4$$

$$(2+1)*2 = 6$$

class ThreeDArray {

public static void main (String [] args) {

```
int arr[3][3][3] = {{ { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } },
    { { 10, 11, 12 }, { 13, 14, 15 }, { 16, 17, 18 } } };
```



Plane no.

①

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

	0	1	2
0	10	11	12
1	13	14	15
2	16	17	18

arr[0][2][1]
 ↓ plane
 ↓ row
 ↓ col.

arr[0][1][1]
 ↓ plane

~~rect 41~~ \rightarrow ~~seen - 6 - 7 - 23~~

Page No.:	1
Date:	1/1/2023

null pointer Exception.

class NullpointerDemo {

 public static void main (String [] args) {

 declare & initialize int arr1 [] [] = {{}, {}, {}};

 int arr2 [] [] = new int [2] [];

 System.out.println (arr1.length); // 3
 System.out.println (arr1[0].length); // 0

0	0	0
0	0	0
0	0	0

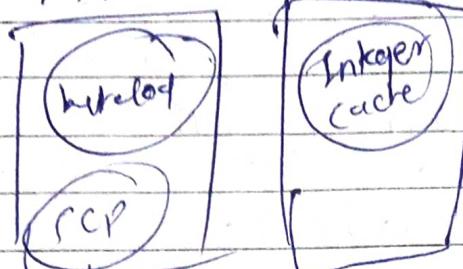
 System.out.println (arr2.length); // 2

 System.out.println (arr2[0].length); // exception

 } \rightarrow Null Pointer Exception
 arr2[0] = 0 0

* String ****

method area Heap



String
constant pool

48 - 0

49 - 1

50 - 2

2 = 90

class StringDemo {

public static void main(String[] args) {

String str1 = "Core2Web"; // SCP

identity hashCode
gives unique ID
not address

String str2 = new String("Core2Web"); // Heap

char str3[] = {'C', '2', 'W'}; // Heap

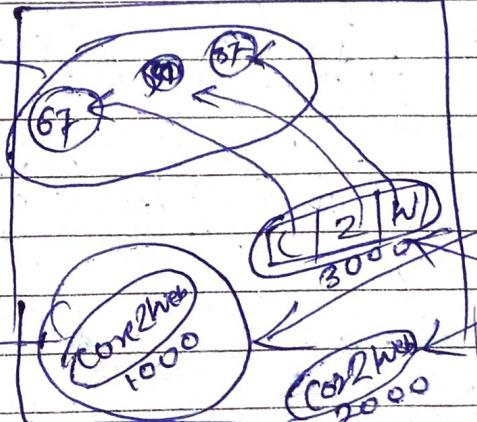
SOP(str1);

SOP(str2);

SOP(str3);

integer
cache

SCP



Stack

1000	str1	new
2000	str2	
3000	str3	

rect 42
5-7-23
seen 6-7-23

Page No. / /
Date / /

Problem 1

String 1 and 02

String
char arr = {'A', 'B', 'C'};

SOP(arr);

String शोक्त असें तर
सब का string मध्य होता जाता.

SOP("Array= " + arr);

{
toString() class
implemented in both
String & array class
so shown as java.lang.Object}

// arr.toString()

Opp => Array = CC 00 716361
address

Problem 2

(P2) identity hashCode feature

Method (असारी समित)

imp statement

valueof(ch) (call chen)

primitive type
datatype

class CacheDemo {

public static void main(String[] args) {

javap -c CacheDemo.class
checked at commandline

char ch = 'A'; char c = 65;
int x = 65; int y = 'A';

SOP(system.identityHashCode(ch));
diff of
INC SOP(system.identityHashCode(x));

A-Z = 65-90
a-z = 97-122
0-9 = 48-57
ASCII values

but char ch and c and int x & y having
same identityHashCode.

~~int arr1 = {10, 200, 300};~~

~~Integer arr2 = {10, 200, 300};~~

→ ~~int~~ ~~String~~ double print ~~arr1~~
 ↗ identity hashCode same ~~arr1~~.

→ ~~arr2~~ ~~object~~ ~~arr2~~.

→ ~~int~~ ~~double~~ print ~~arr2~~ ↗
 ↗ identity hashCode.

print ~~arr1~~ ↗ 200 & 300 ↗
 ↗ identity hashCode ~~arr1~~ ↗ ~~arr2~~

String str1 = "kankha";

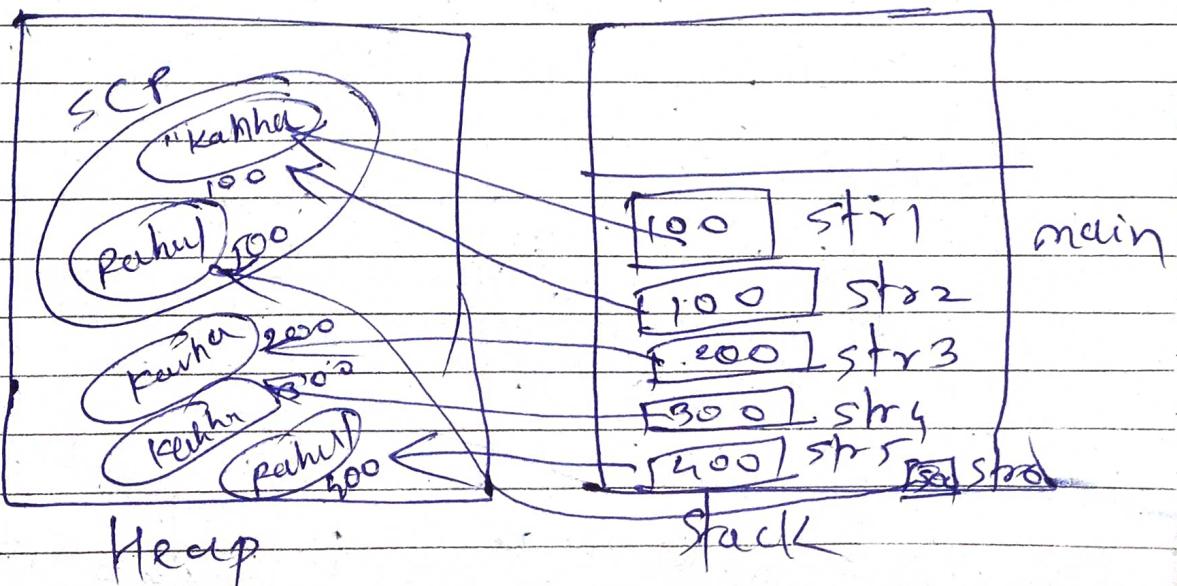
String str2 = "kankha";

String str3 = new String("kankha");

String str4 = new String("kankha");

String str5 = new String("Rahul");

String str6 = "Rahul";



mutable → changeable

Page No. _____

Date _____

Java string is immutable → not changeable
(constant)

to check method

→ javap java.lang.String

```
class StringDemo {  
    public static void main(String[] args){  
        String str1 = "Dnyaneshwar";  
        String str2 = "Pingale";  
        System.out.println(str1 + str2);  
    }  
}
```

System.out.println(str1 + str2);

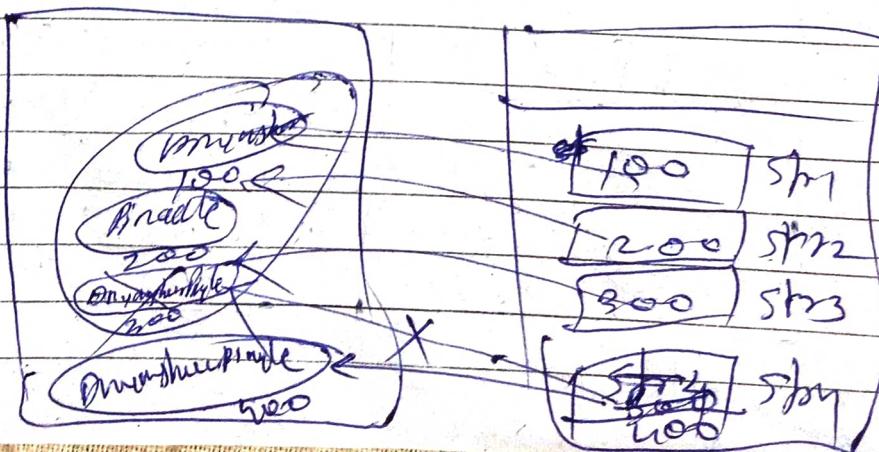
~~String str1 = "Dnyaneshwar";~~

String str3 = "DnyaneshwarPingale";

String str4 = str1 + str2;

System.out.println(str3);

System.out.println(str4);



~~Shoe Dog book~~ } → Nyke owner
311+100%

Page No.:	/ /
Date:	/ /

```
String str1 = "Shashi";
String str2 = "Baigal";
```

str1.concat(str2);

`SOP(str);`

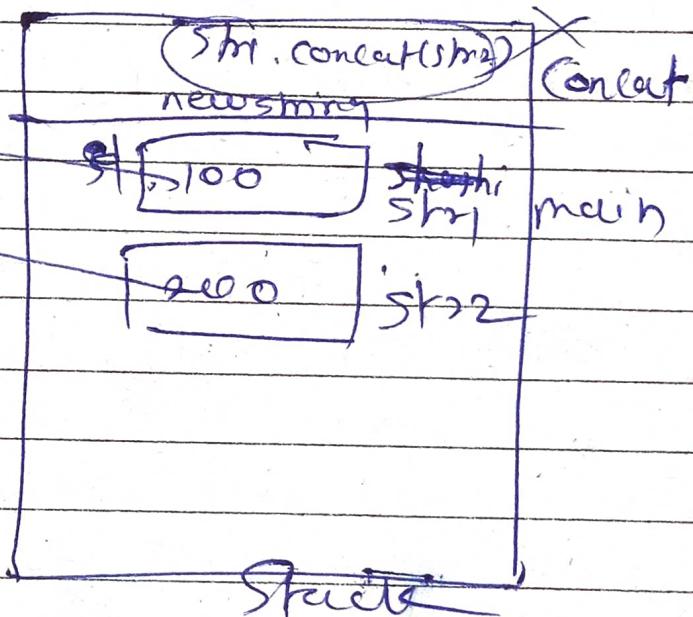
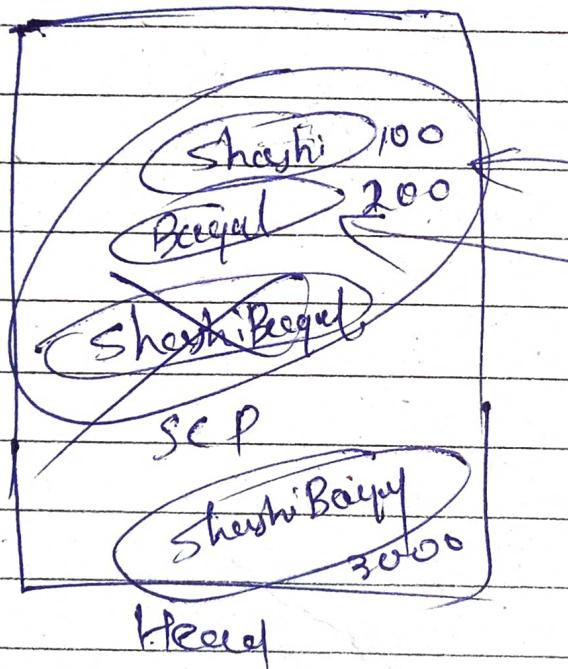
```
sop(str1);
```

~~sop(s)~~

str1.concat(str2);

SOP (Str1)

SOP (star);



leet 43: String in Java 03

seen 9-4-23

Sunday
Date:

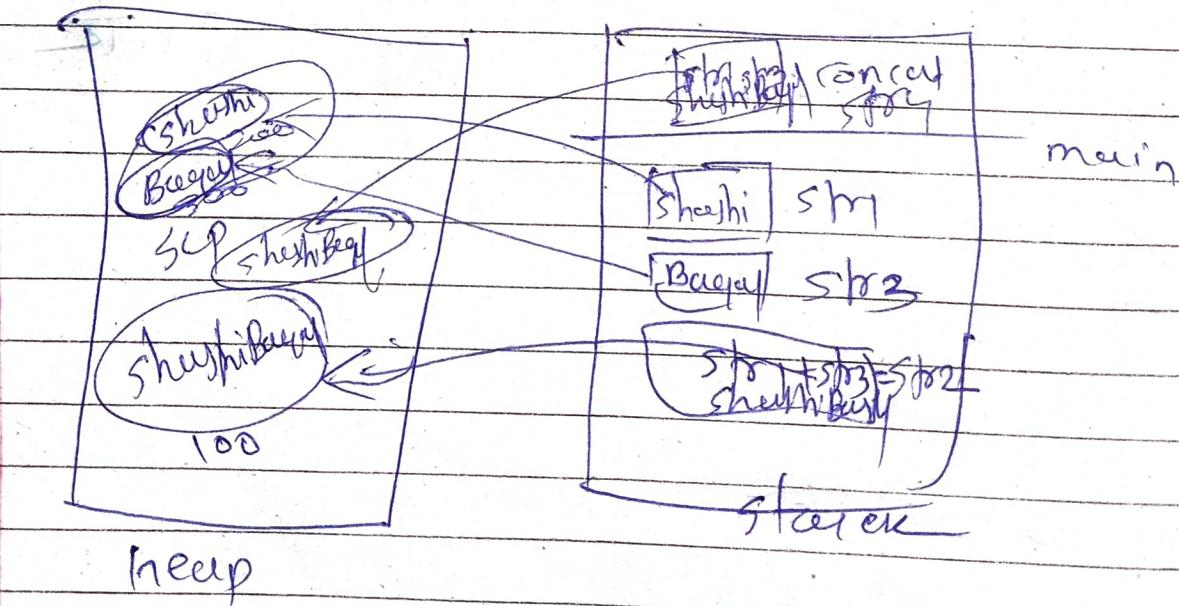
→ quiz & explanation ~~allgemein~~ ~~hier~~ from E1.

`==` operator identity hashCode compare
check. only for string

(class String Demo)

direct same
reference
performance
cost
memory
management
String str1 = "Shashi";
String str3 = "Boyal";
String str2 = str1 + str3;
String str4 = str1.concat(str3);
System.out.println(str2);
System.out.println(str4);

O/P with diagram



HashCode

source code of
Object class
Search
in google

```
class HashCodeDemo{
```

```
    public static void main(String[] args)
```

```
        String str1 = "shashi";
```

```
        String str2 = new String ("shash");
```

```
        String str3 = "shushi";
```

```
        String str4 = new String ("Shasti");
```

```
        System.out.println(str1.hashCode());
```

```
        System.out.println(str2.hashCode());
```

```
        System.out.println(str3.hashCode());
```

```
        System.out.println(str4.hashCode());
```

Output

-1899698008

Capital & lower case conversion
code

$S + 32 \Rightarrow$ lower case 's'

'A' \Rightarrow 'a'

$65 - 97 \Rightarrow 32$

methods

① concat

```
public String concat(String str);
```

description:

concatenate string to this string i.e.

Another string is concatenated with the
first string

→ Implements new array of character whose
length is sum of str1.length and str2.
length + p

Parameters: String
return type: String

class concatDemo {

public static void main (String [] args) {

String str1 = "core2";

String str2 = "web";

String str3 = str1.concat(str2);

~~System.out.println~~

System.out.println(str3);

}

Output: core2web

{}

② method 2: public int length();

Description: It returns the no of char contained in given string.

parameters: No parameter

return Type: Integer

class lengthDemo {

public static void main (String [] args) {

String str1 = "core2web";

System.out.println(str1.length());

myLength()

Q3) method 3:

public char charAt (int index);

description: if returns character located at specified index within given string.

parameters: int index(index)
return type: character.

class CharATDemo

public static void main (String args) {

String str = "Core2Web";

Sop(str.charAt()); // 2

Sop(str.charAt(0)); // C

Sop(str.charAt(8)); // StringIndexoutof
BoundaryException

39

method 4:

public int compareto (String str2);

Description: it compares the str1 and str2 (case sensitive); if both the strings are equal it returns 0 otherwise returns the comparison.
Ex. str1.compareto(str2).

parameters: string (Second String).

return type: integer.

→ ~~unfilled~~ StringBuffer str2 ~~anumit str1~~

class CompareToDemo{

public static void main(String[] args){

String str1 = "Ashish"; ~~or Ashish~~

String str2 = "ashish"; ~~or Ashish~~

Scop(str1.compareTo(str2))

Top ➔ 32

it return difference between two string

(methods):

public int compareToIgnoreCase (String str);

Description: It compare str1 and str2 (case insensitive).

parameter: string

return type: integer.

```
class CompanyIgnoref  
public static void main (String[] args){
```

```
String str1 = "SHASHI";
```

```
String str2 = "shashi";
```

```
Sop (str1, compareToIgnoreCase(str2));
```

or

```
str1 = "SHASHI";
```

```
str2 = "shashikant";
```

```
Sop (str1, compareToIgnoreCase(str2));
```

O/P - 4.

method 6:

method: public boolean equals(Object anObject);

description: predicate with compare and object to this. This is true only for strings with the same character sequences returns true if and object is semantically equal to this.

Parameter Object(anObject)

return type boolean.

```
class EqualsDemo{
```

```
public static void main (String[] args){
```

```
String str1 = "shashi";
```

```
String str2 = new String ("shashi");
```

```
Sop (str1.equals(str2)); //
```

→ compare content → true,

Key 44:

Date: / /

String in Java

seen at 9-7-23

* Conversion String to charArray:

```
class TochararrayDemo {
    public static void main(String[] args) {
        String str1 = "shahibagyal";
        char arr[] = str1.toCharArray();
        for (int i = 0; i < arr.length; i++) {
            System.out.println(arr[i]);
        }
    }
}
```

* my method to find length of string.

```
static int mystrlen(String str) {
    char arr[] = str.toCharArray();
    int count = 0;
    for (int i = 0; i < arr.length; i++) {
        count++;
    }
}
```

return count;

```
public static void main(String[] args) {
    String str1 = "shashibagyal";
    int len = mystrlen(str1);
    System.out.println(len);
}
```

import java.util.Scanner;

scanner sc = new Scanner(System.in)

String str1 = sc.next();

String str2 = sc.next();

if (mystrlen(str1) == mystrlen(str2)) {

System.out.println("both are equal");

}

else {

System.out.println("not equal");

* Compare to implement equality rule.

public boolean

method: equalsIgnoreCase (Comparing another String)

Description: Compares a String to this string ignoring case.

parameters: String (str2)

return type: boolean.

Method 4

public int indexof (int ch, int fromIndex);

description:

finds the first instance of the character in the given string.

parameters: character (ch to find), Integer (index to start the search)

return Type: Integer.

class IndexofDemo {

public static void main (String args) {

```

String str1 = "Shashi";
System.out.println(str1.indexOf('h', 0));
System.out.println(str1.indexOf('h', 1));
System.out.println(str1.indexOf('h', 2));

```

O/P :-
 1
 1
 4

Shashi
 0 1 2 3 4 5

Method 5lastIndexof (int ch, int ^{upto} fromIndex);

description: finds the last instance of the character in the given string

parameters: character (ch to find), Integer (index to start the search)

return Type: Integer.

```
class lastIndexofDemo {
```

```
public static void main (String [] args) {
```

```
String str1 = "Shashi";  
System.out.println(lastIndexof('h', 0));
```

method 12:

```
public String replace (char oldchar, char  
newchar);
```

description: replaces every instance of a character in the given string with a new character.

parameters: characters (old character), character (new character).

return type: string,

```
class replaceDemo {
```

```
public static void main (String [] args) {
```

```
String str1 = "Shashi";
```

```
System.out.println(str1.replace('s', 'a'));
```

method 13 :

public String substring (int index);

description : Creates a substring of the given String starting at a specified index and ending at the end of given String.

parameters: Integer (index of the String),

return type: String,

class substringDemo {

public static void main (String [] args) {

String str = "Core2Web Tech";
 ^ 1 2 3

Sop (str.substring (5));
Sop (→ (0, 3))

OP / Web Tech.
→ Core

string → immutable
StringBuffer → mutable

Page No. / /

Date: / /

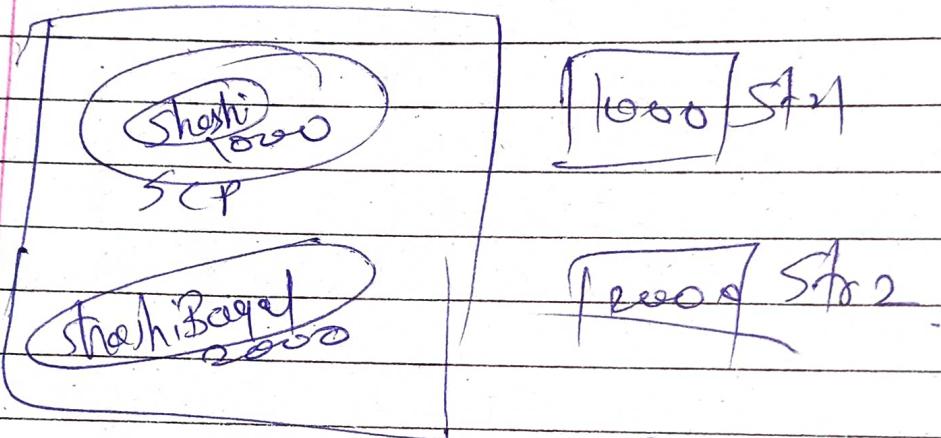
StringBuffer *

string str1 = "shashi";

string str2 = str1.concat(" Bagal");

sop(str1);

sop(str2);

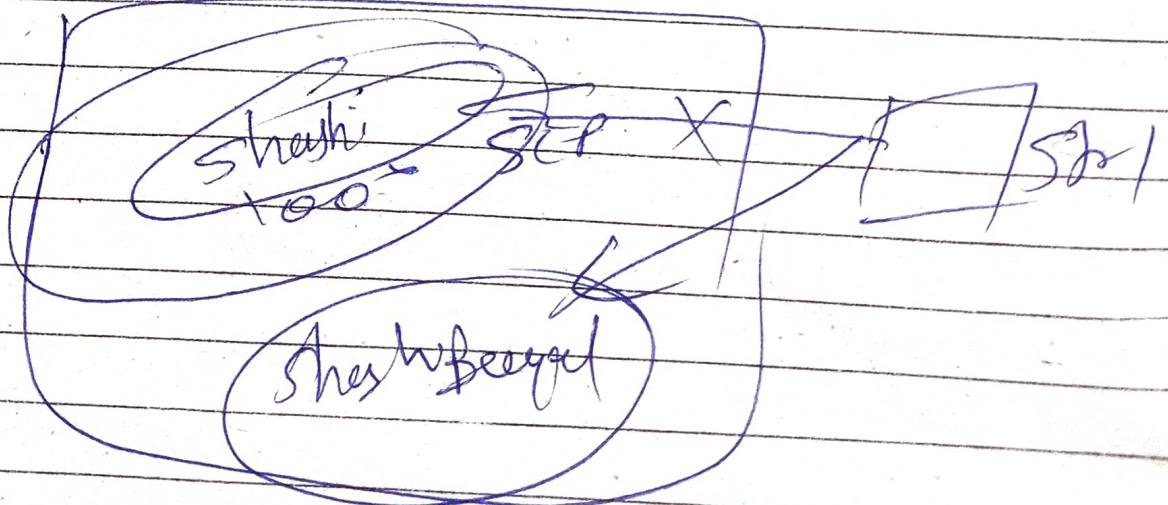


or

string str1 = "shashi";

str1 = str1.concat(" Bagal");

sop(str1);



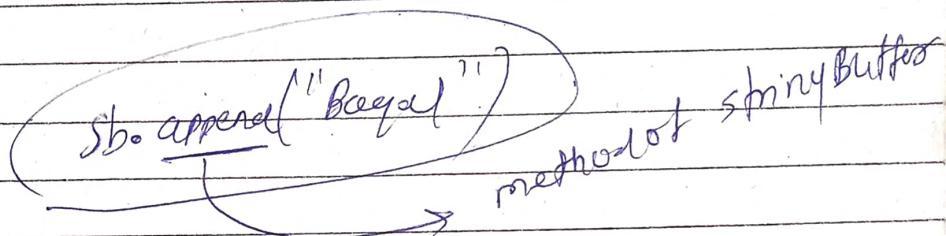
String class ~~is immutable~~ ~~is thread safe~~ ~~is synchronized~~
 String change ~~is not thread safe~~ ~~is not synchronized~~

Page No.:	1
Date:	1/1/2023

String Buffer class ~~is mutable~~ ~~is not thread safe~~ ~~is not synchronized~~
 String ~~is mutable~~ ~~is not thread safe~~ ~~is not synchronized~~ change

String Buffer sb = new String Buffer ("Sheh")

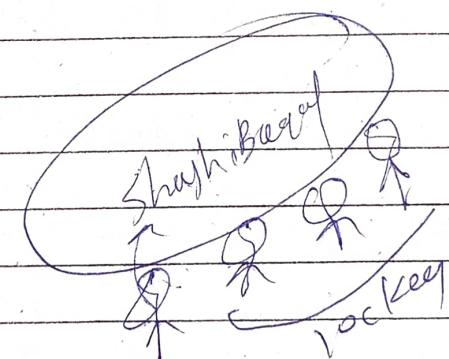
StringBuffer SCP at ~~not~~ ~~not~~ ~~not~~
 String at ~~not~~ SCP at ~~not~~ ~~not~~



→ ~~not thread safe~~ cost ~~more~~

so what

String Buffer ~~permits~~ methods synchronized



When multiple
threads come
then it
will only
work one
thread

Class	String	String Buffer	String Builder
	immutable	mutable	mutable
	synchronized	asynchronized	
	one thread work	multiple threads work	

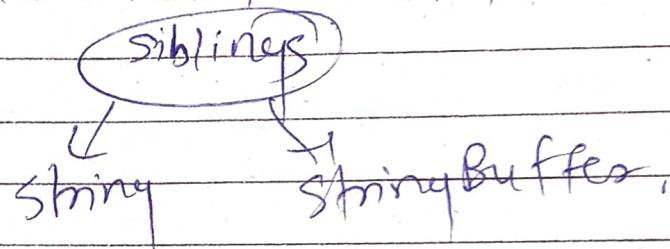
Date: _____

StringBuffer str1 = "Shashi";

↑
3121) string ~~not extendable~~
~~not changeable~~
~~not reusable~~
not ~~SCP~~ ~~OK~~ ~~thin~~
not ~~reusable~~.

error: incompatible type

example: ~~String~~ ~~String~~ example



class StringBufferDemo{

public static void main(String[] args){

StringBuffer str1 = new StringBuffer("Shashi");

Sop(system.identityHashCode(str1));

str1.append("Bugal");

Sop(str1);

Sop(system.identityHashCode(str1));

Sop(str1.capacity());

String Buffer -> default capacity 16

~~StringBuffer & StringBuilder~~

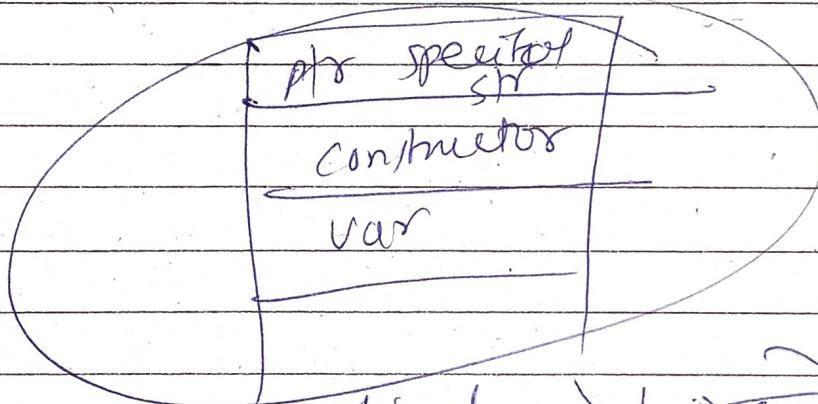
~~Lesson 3~~

seen 10/7/23
Page No.: 1/1

~~String & StringBuffer~~ since 1.0

2nd year of 3rd year student \rightarrow TA (Teacher Assistant)
31/10 in scalar

~~String~~ container \rightarrow ~~dynamic~~ type-~~u~~
like ~~data stru~~, wallet, bag



~~data~~ object \rightarrow ~~doest~~ heavy
operation

~~Using~~ StringBuffer

~~StringBuilders~~ 1.5 ~~use~~ 31/10/23

~~String~~ ~~construct~~ ~~struc~~ ~~data~~ ~~in~~
server area ~~store~~ ~~data~~

StringBuffer (lang package) Date: 1/1

String Buffer sb = new StringBuffer();

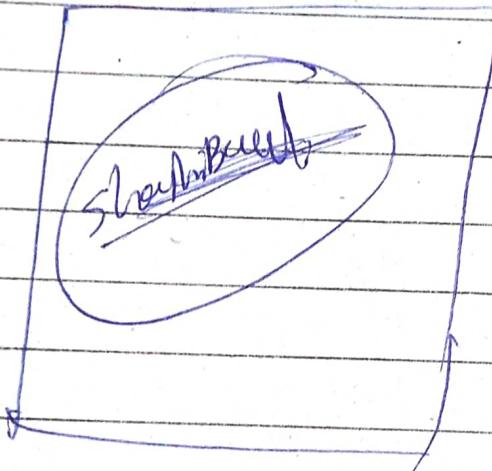
sb.capacity(); \leftarrow 16 ✓

sop(sb); \leftarrow

sb.append("shaashi");

sb.capacity(); \leftarrow 16 ✓

sb.append("Bugal");



sop(sb.capacity());

sop(sb);

sb.append("ore2web");

sop(sb.capacity());

sop(sh);

~~lect 46~~
seen at 12-7-23 Page No. ~~StringBuffer and StringBuider 02~~

→ Example → ~~this~~ city ~~don't~~ ~~plots~~

EcoTech → ~~any~~ product

↳ ~~any~~ virtual machine.

heap size ~~any~~ ~~any~~ ~~any~~.

↳ we can request to JVM.

~~for increase~~ size = formula in StringBuffer
 $(\text{current capacity} + 1) \times 2$

```
StringBuffer sb = new StringBuffer();  
sb.append("Biencaps");  
sb.append("Core2Web");  
System.out.println(sb);  
System.out.println(sb.capacity());
```

fill ratio (concept \Rightarrow there is no need now)

but
at start 75% $\frac{\text{buffer size}}{\text{capacity}}$
असत दर से 75% पूरा
capacity होने पर)

sir only talk about

vector concept \Rightarrow ~~to~~ ~~on~~ ~~not~~.

```
sb.append("Incubator");
```

```
System.out.println(sb);
```

```
System.out.println(sb.capacity());
```

ensureCapacity or initialCapacity

array of size 10 not 30 25 20 etc
+ test cases.

StringBuffer sb = new StringBuffer(100);
sb.append("Biencaps");
sb.append("Core2Web");
sop(sb);
sop(sb.capacity());
sb.append("Incubator");
sop(sb) + sop(sb.capacity);

① StringBuffer sb = new StringBuffer();
② → (100);
③ → ("sheikh")

↑
at line constructor m1
represent charin.

object of constructor m1 represent.

→ यह अस क्षण जैसे CPU का प्रयोग करें।
→ INT calculations करता है। (CPU
use INT or instructions. INT in
calculations करता है।)

(Instruction call करता है।
INT in.)

```
class SBDemo {
```

```
public static void main (String [] args) {
```

```
String str1 = "Shashi";
```

```
String str2 = new String ("Bagal");
```

```
StringBuffer str3 = new StringBuffer ("Rakesh");
```

```
String str4 = str1.append(str3);
```

```
SOP (str1);
```

```
SOP (str2);
```

```
SOP (str3);
```

```
SOP (str4);
```

~~OP:-~~ cannot find symbol

→ String ~~as class~~ has append method

→ method of String

concat and concat.

method of
StringBuffer
class

```
StringBuffer str4 = str3.append(str1);
```

→ No error

~~error~~

No need

return
newString Buffer
(concatenation)
object

to create

new StringBuffer

like this.

append is powerful than concat

Project No. 1
Java String

java append

Ques

String str1 = "shashi";

String str2 = new String("Begal");

StringBuffer str3 = new StringBuffer("core2web");

StringBuffer str4 = str3.append(str1);

sop(str1);

sop(str2);

sop(str3);

sop(str4);

String str1 = "shashi";

String str2 = new String("Begal");

StringBuffer str3 = new StringBuffer("core2web");

String str4 = str1.concat(str3);

error
incompatible
type

StringBuffer str5 = str3.append(str2);

sop(str1); → shashi

sop(str2); → Begal

sop(str3); → core2webBegal

sop(str4); → shashi(core2web)

sop(str5); → core2webshash
Begal

Concat(~~of~~ string class -)

method static static final

String String set parameter

String StringBuffer object

String str1 = new String("Shuhi");

String str2 = new String("Bagel");

StringBuffer str3 = new StringBuffer();

str1.concat(str2);

str3.append(str2);

sop(str1); // Shuhi.

sop(str2); // Bagel

sop(str3); // corelwehBagel.

append



* method :

public synchronized void StringBuffer(String str);

* method : insert()

→ StringBuffer insert (int offset,
String str);

insert the <code> String </code> argument int
this <code> StringBuffer </code>.

Code

```
class SBDemo {  
    public static void main (String[] args) {  
        String Buffer sb = new Stringbuffer  
        ("sheesh(oreWeb)");  
        System.out.println (sb);  
        sb.insert (6, "Beqa1");  
        sop (sb);  
    }  
}
```

③ method : Delete

```
StringBuffer (int start, int end);
```

- Delete character from this

```
<code> StringBuffer </code>
```

```
<code> delete (10)/12)</code> will delete
```

10 and 11, but not 12.

- If is hamless for end to be larger
than length().

```
class SBDemo {
```

```
public static void main (String[] args) {
```

```
    Stringbuffer sb = new Stringbuffer ("oreWeb");  
    sb.delete (2,7);
```

```
    sop (sb);  
}
```

O/P:- Cob

④ Reverse

\$ \$ \$ \$

StringBuffer reverse();

- reverse the characters in this
StringBuffer.

- The same sequence of characters exist,
but in the reverse index order

class ReverseDemo {

① public static void main(String args) {

StringBuffer sb = new StringBuffer
("Shashi");

Sop(sb.reverse());

}

②

String str = "core2web";

StringBuffer sb2 = new StringBuffer(str);

str1 = sb2.reverse().~~to~~ String();

Sop(str1)

~~toString~~

method of object class

object class CT ~~का प्रियोनीय~~ parent class

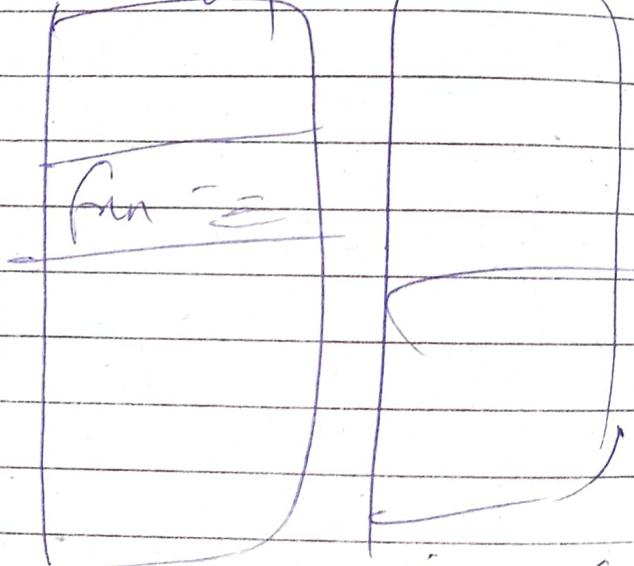
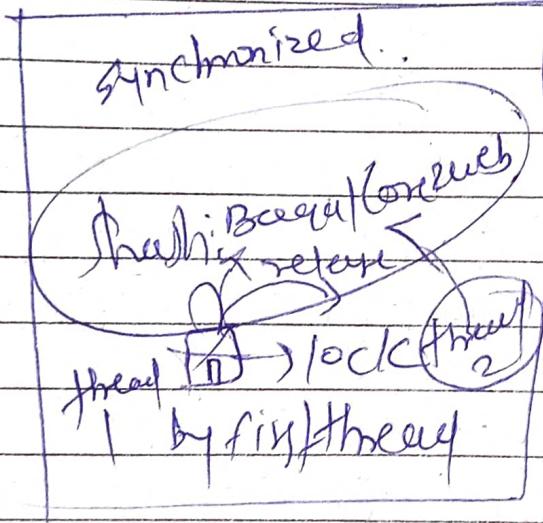
Page No.

Synchronized :-

~~before it~~

जैविक thread में कोई तबादी नहीं
stack में भी

आतापर्यंत 31/01/2023 को 2 thread औपर लिए
Baeat foreword



java stack java stack

string builder केवल at a time multiple
thread 23 जून 2023

Introduction to OOPclasses & object

→ Networking

→ hacking

→ ~~function~~ → ~~data~~ idenit.→ Zomato →

cloud kitchen chain only zomato

→ website ~~function~~

web app → LinkedIn, Instagram | dynamic

website → wikipedia (static)

mobile apps →

→ flutter ~~function~~ web or app ~~function~~ios or, android or ~~function~~→ flutter ~~function~~ oop ~~function~~ language ~~function~~

Dart

↳ exactly same as java

→ 2017 mit flutter ~~function~~→ check source code of StringBuffer

search on google.

→ java ~~function~~ destructor ~~function~~ ~~function~~Garbage collector ~~function~~.→ java ~~function~~ constructor ~~function~~

20%

30%

mercy
m/dy
Date: 8/27/2027
code chmn.

locat static → class variable

non static → instance variable

→ ~~जावा का हेप ऑफियल संस्करण क्या है?~~

→ constructor and class ~~से~~ नहीं same
31<10

→ ~~मिले special method फॉनिंग
return type भी है~~

constructor मिले call छोड़ते हैं <init> जावा की
→ दो रूप हैं

method java /lang / object .<init> "();"

fr
object class से
constructor मिले call
one

→ constructor तो इन छोड़ते हैं आमतौर पर
→ instance variable initialize करते हैं
store करते हैं छोड़ते हैं आमतौर पर

→ instance variable नस्तीक तरीके पर
constructor के लिए

```
class Demo {
```

```
}
```

byte code:

```
class Demo {
```

```
    Demo();  
    Code :
```

```
0: aload_0
```

```
1: invokespecial #1           // method java  
/lang/Object.<init>:()V
```

```
}
```

class Demo {

Demo();
 super();

?

This call

From

~~all~~ ~~first~~ parent class

~~all~~ constructor

call

→ java by default inheritance follow
top

↳ java bottom uses top uses
uses object class defn.

javap -c java.lang.Object

किंतु जैसा लिखा है

instance variable m1 एवं constructor
Head असेहो.

→ compiler ने इनमें से किसी नहीं constructor
add किया.

⇒ Constructor m1 return type नहीं

→ object मात्र का है तर एक constructor m1
class का नहीं होता.

constructor m1 का
नाम नहीं होता वर्तमान
variable m1 का name होता है.

That's why

object is must for accessing
this nonstatic variable.

class Demo {
 int age = 10;
 int jenoNo = 18;
 static void fun() {
 int x = 10;
 }
}

Byte code

class Demo {
 int age; int jenoNo;
 Demo() {
 super();
 }

super(); invokespecial

super(); // invoke
special

1000

age = 10; // bipush
jenoNo = 18; // bipush

call the
parent
class
constructor

bipush

in bytecode

void fun() {

x = 20; // bipush

static instance or global variable ~~is~~
byte code ~~has~~ static block ~~has~~
~~has~~

class Demo {

 int age = 10
 static int field jernno = 18

}

byte code

Demo();

 ; invoke special
 bipush 10

statics { ; }

bipush 18