String Buffer.

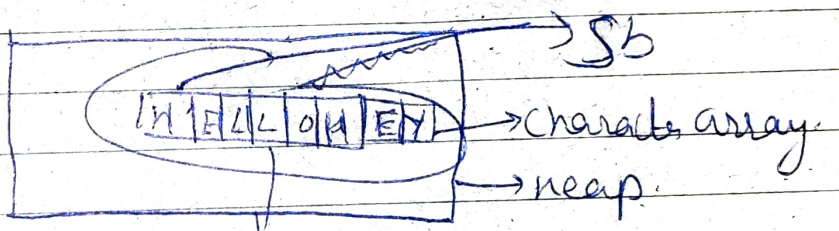
⑥ Never Create a object of StringBuffer via literal

~~StringBuffer Sb = "Hello"~~

Making Object of StringBuffer via
- ① Default Constructor
  - ↳ can be constructed via Constructor.

1 StringBuffer Sb = new StringBuffer( )
↳ default Constructor of StringBuffer.

- String & StringBuffer have many common method but these common methods dont come from a common parent but instead they are redefined for both StringBuffer




```
Sop(sb.capacity());   → 16
Sop(sb.length());     → 0
sb.append("Hello");
sb.append("Hey");
```

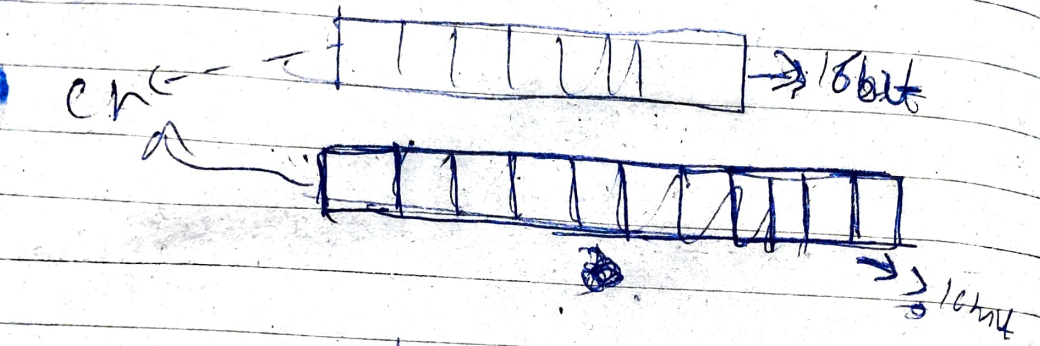
→ method to print size of array.

Que →
- But Since Array is has initially 16 bit only, so if we add > 16 bit then what happens?

Ans → Then there should be Array Index Out of bound Except should occur but it doesn't, because of follows:

This is achieved by creating a new array of larger size, inserting values in it and then passing its ref. to the existing Ref. Variable, & older array is collected by GC.

Q_{ries}



This is dynamic growing of array. using new array

New arrays cap. capacity is given by

(Initial Capacity + 1) * 2

Que → But what if we append in the StringBuffer for 2nd time we insert a 90 character long String?

Ans → It will make the capacity as 90 and insert 90 characters in that

2

(2) StringBuffer (int capacity);

user input initial capacity

Eg

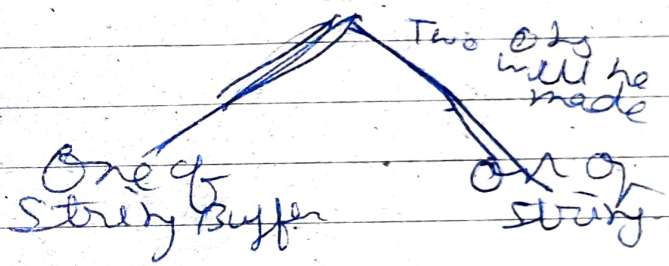
StringBuffer Sb = new StringBuffer(20);

if ≥ 'int capacity' characters is inserted at a time

then its capacity becomes [(int capacity + 1) * 2] like previous case

(3)

StringBuffer Sb = new StringBuffer ("Hello");



Initial length = 16 + length of "Hello"

our given string.

Initial length = 16 + length of (our given string)

~~Modification Methods~~

Example → Creating StringBuffer via

```
class StringB
{
  psvm(String... s)
  {
    ~~String Sb~~
    StringBuffer sb = new ~~new~~
                      StringBuffer();

    //StringBuffer Sb = new StringBuffer(20);
    //  "         Sb =  "      " ("duct");

    Sopln(Sb.length());
    Sopln(Sb.capacity());
    StringBuffer Sb1 = sb.append
             ("aa.........a");
              17 times / 90 times
    Sopln("after append");

    Sopln(Sb.length());
    Sopln(Sb.capacity());
```

```
sb.append("a");
System.out.println("after second append");
So pln(sb.capacity());
}
}
```

Modifier Methods

```
class StringBInsert
{
psvm(String...s)
{
StringBuffer sb = new StringBuffer("I Java");
Sopln(sb);
So pln(sb.length());
Sopln(sb.capacity());
```

changes in same string so, no need to create new

```
StringBuffer sb1 = sb.insert(2,"like");
```

insert at this index position

```
String Sopln(sb)
sb1 = sb.replace(3,6,"love");
```

from this indx — this indx

```
Sopln(sb);
```

```
StringBuffer sb1 = sb.delete(2,6);
System.out.println(sb);

StringBuffer sb1 = sb.reverse();
System.out.println(sb);
if (sb1 == sb)
System.out.println("one Object");
}
```