Strings ⇒ array of characters

' ' → character literal

String a = "Geekster";

Memory → space minimize

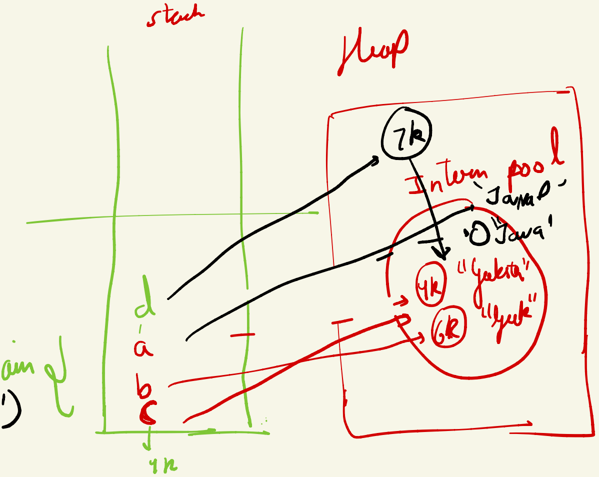
String a = "Geekster';

String b = "Geek";

String c = "Geekster";

String d = new String("Geekster")

main



a = "java"

a = a + 'D';

a = a + "DSA";

why? → because of intern pool

1) Strings are immutable in Java. → you can not change anything once defined.

String a = "Java"

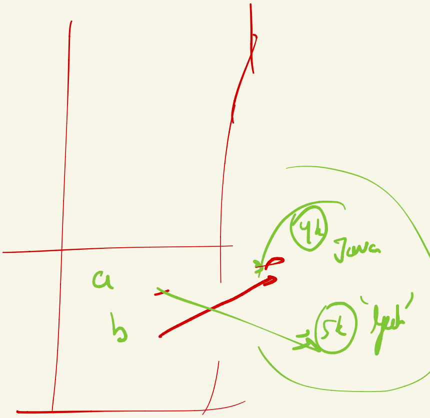
String b = a;

a = "Geek"; print(a), print(b) → Java

Geek ← print(a)

b = a; print(a), print(b) → Geek

Geek ← print(a)



```
String a= "Java";
string b= "Java";
String c= new String ("Java");

if (a==b){
    print ("a is equal to b");
}
if ( a==c) {
    print ("a is equal to c");
}
```

== ↓ addresses check →

.equals ↓ address check ↳

character by character check

```
if (a.equals(c)){
    print ("a is equal to c");
}
```

How to take Input

```
String a= scn.nextLine();
```

String → array of characters

```
              0 1 2 3  4 5 6 7
string a= "   g e e k  s t e r"

char ch = a.charAt(3);   // ch=='k'
```

length of string :

```
String str = scn.nextLine();
int len = str.length();
```

Ques Take a input string. Count the number of vowels in that string.

eg. ⇒ geekipu

output → 4

[e, e, i, u]

Ques Print the string in reverse order.

0 1 2 3 4 5 6

eg → "geekipu"

output → "upikeeg"

```
string a = "nano"
for (int i = 0; i < 10; i++) {
    a = a + i  → O(n)
}
```

a = nano

a = nano1

a = nano12

a = nano

true ← → false

if (isVowel(ch)) {

→

}

true false

if (____)

(isVowel (ch) == false) → true , false

(isVowel(ch)= = true

if (false)

true , false

if (isVowel (ch))