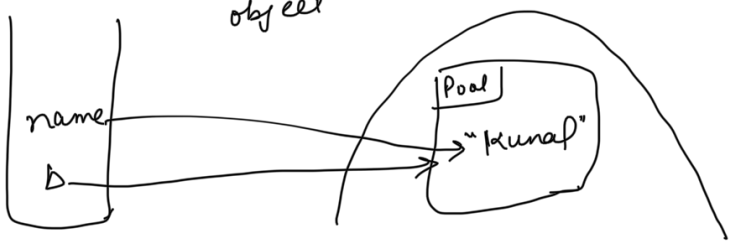
String.

String name = "Kunal"
object

String b = "Kunal"

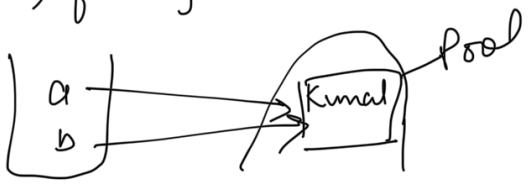


Same Strings are not recreated in pool heap.

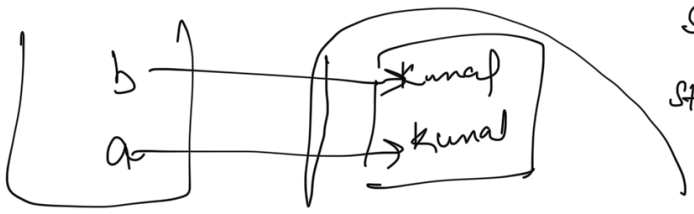
String objects are immutable due to security. If one can change a string then it will be changed for other variables too.

String a = "Kunal"
a = "Natasha" ← do not change Kunal creates a new object named natasha. Kunal will be deleted at time of Garbage collection if it is not being pointed by any other reference variable.

a == b (true) if object is same



a == b (false) if object are not same



String b = "Kunal";
String a = new String("Kunal");

forced to create new object no matter what

→ Comparison a.equals(b)

→ PrintStream is class which has method println which is overloaded as per arguments.

→ System.out.println(Object);
calls Object.toString(). Returns a textual representation of object.
↳ this can be overridden.
for eg Arrays.toString()

→ System.out.println(56) This is not int. It uses Integer Wrapper class and calls to String internally

→ Pretty Printer %15d, %15s, %0.2f, %-15s.
Placeholder in Java
↑ left justified

→ When an int is concatenated to a string, it is converted to wrapper class Integer and concatenated as string.
"a" + 1 = "a1"
= "a" + "1" only

All object in sout will call toString.

→ operator '+' is only defined for primitives, or if any of value is of type string.

→ Operator overloading is not supported.

→ Concatenating to string will create a new string everytime a change is made.
↳ a loop changing a string will create multiple items