

થોચને યે કાઢું સિહ્યો હું પણનાઓ કે શહર
ચઢના રીતી જરૂરી હું માંવેલ પણે કે ખિએ !!

Page No.: 1 / 1

④ Strings

- A String is a sequence of characters.
- In Java, String is an object.
- Java Platform Provides the "String" class to create & manipulate strings.

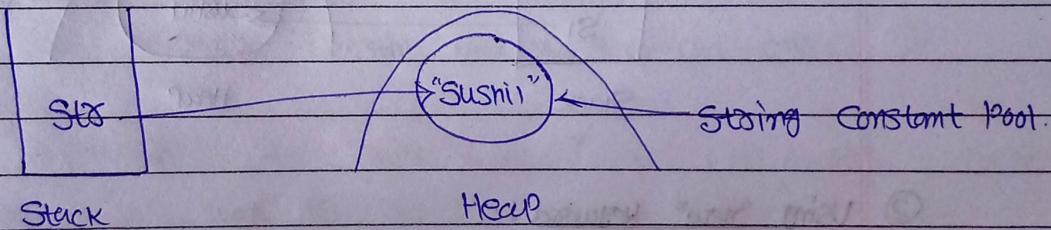
* Creation of Strings in Java *

We can create a string in 2 ways.

- By String Literal.
- By "new" keyword.

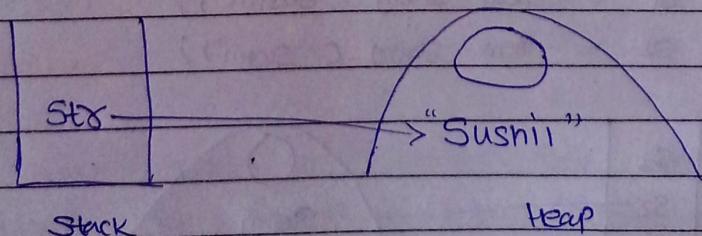
① String Using String Literal :- In this way we can directly assign a string literal to a string reference.

Ex:- String str = "Sushi";



② String Using "new" Keyword :- We can dynamically create a string by using "new" keyword.

Ex:- String str = new String ("Sushi");



* Concept Of String Constant Pool *

It is a special area in java heap memory where string literals are stored. Whenever a string literal is created, compiler checks the

* Complete process of String creation *

① Using String Literal :-

Whenever a String literal is created :

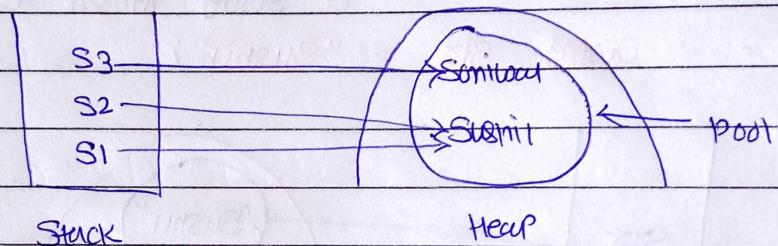
- JVM will check the "String Constant Pool" first.
- If string already exists; A ref. to be pooled instance is returned.
- If string doesn't exist; A new object will be created.

Ex:-

String S1 = "Sushil";

String S2 = "Sushil";

String S3 = "Sonitabud";



② Using "new" keyword :-

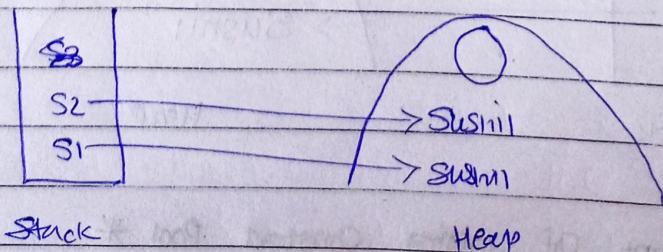
Whenever we create a String using new keyword :-

- JVM creates a new String object in heap (Non-Pool).
- From this we can create some objects.

Ex:-

String S1 = new String ("Sushil");

String S2 = new String ("Sushil");



* Immutable Behaviour of Strings *

String Objects are immutable in Java which means After once creating a String Object, we can't change / modify this, But we can create a new one.

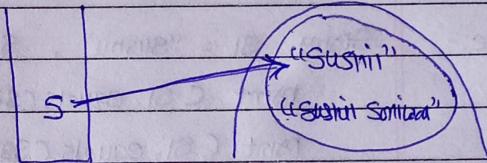
* Why Immutability :- ① Security ② Heap Space ③ Thread Safe.

* Example of String Immutability :-

Ex:- ① String s = "Sushi";

s.concat("Sonika");

Print(s);



Output:- Sushi.

Why:- In above example original object isn't changed, a new one is created but ref. Variable "s" is still pointing to old one.

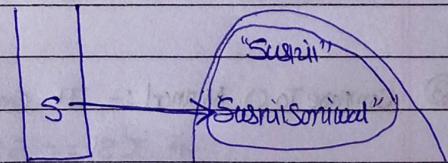
Ques:- How can we refer s to SushikSonika ?

Sol:- We can explicitly assign s to it.

Ex:- ② String s = "Sushi";

s = s.concat("Sonika");

Print(s);



Output:- SushikSonika.

Why:- In the above example a completely new object is created.

The previously created one will be automatically destroyed by Java Garbage Collector.

* Comparison of Strings *

We can compare two strings on the basis of their content and references.

We are having 3 ways for comparison.

- (1) equals() Method :- It compares the original content or value.
- (a) • equals() :- Compares specified object.
- (b) • equalsIgnoreCase() :- Compares string but ignore case.

Example:-

String s1 = "Sushi" , s2 = new String ("Sushi") , s3 = "SUSHI"
Print (s1.equals(s2)); || True
Print (s1.equals(s3)); || False
Print (s1.equalsIgnoreCase(s3)); || True

- (2) Comparator(==) Method :- It compares references not values.

Example:-

String s1 = "Sushi" , s2 = "Sushi" , s3 = new String ("Sushi")
Print (s1 == s2); || True
Print (s1 == s3); || False
Print (s2 == s3); || False

- (3) CompareTo() Method :- It compares values lexicographically.

if (s1 == s2) \Rightarrow return 0;
if (s1 > s2) \Rightarrow +ive;
if (s1 < s2) \Rightarrow -ive;

Example:-

String s1 = "Sushi" , s2 = "Somodal" , s3 = new String ("Sushi")
Print (s1.compareTo(s2)); || -6
Print (s1.compareTo(s3)); || 0
Print (s2.compareTo(s3)); || 6

* Concatenation of Strings *

① By Using '' + ' Operator :- It's used to add two strings.

Ex:- String s1 = "Sushi" , s2 = "Soniwal" ;
Print (s1 + s2) ;

Output:- SushiSoniwal

② By Using "".concat()" Method :- It concatenates the specified string to the end of the current string.

Ex:- Print (s1.concat(s2)) ;

Output:- SushiSoniwal

* StringBuffer class in Java *

It is used to create mutable strings.

① Constructors of StringBuffer class :-

Ex:- StringBuffer sbc1 = new StringBuffer () ; || Size of 16 (Initially).
StringBuffer sbc2 = new StringBuffer ("Sushi") ; || Specified string.
StringBuffer sbc3 = new StringBuffer (20) ; || Specified size.

② Methods of StringBuffer class :-

Ex:- StringBuffer sb = new StringBuffer ("Sushi") ;
sb.append ("Java") ; || SushiJava
sb.insert (2, "C++") ; || SucppSushiJava
sb.replace (1, 3, "Python") ; || SPythonnppSushiJava
sb.delete (1, 3) ; || ShapppSushiJava
sb.reverse () ; || avaIIihSppnohs