JavaNotes

****

Prepared by Uday Pawar

**String**

**-> String is pre-defined final class present in java.lang package.**

**-> String Objects Immutable in Nature.**

**-> String is a Collection/Set of Characters.**

**-> String is also a Non-Primitive Datatype.**

**-> String implements Serializable, Comparable, CharSequence**

**-> String Objects can be created in 2 ways.**

**1. literal (" ") double quotations**

**2. using new Operator**

**-------------------------**

**1. String s = "java";**

**2. String s = new String("java");**

**1.**

**package** jspiders;

**public** **class** Example {

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

String s1 = "JAVA";

String s2 = **new** String("JAVA");

}

}

**-> String Objects are stored inside a memory location called as String pool.**

**-> String Pool is further divided into 2 types**

**1. Constant Pool**

**2. Non-Constant Pool**

**-> Literal Objects are stored inside constant pool and constant pool does not allow duplicates.**

**-> String Objects created using new operator are stored inside Non-Constant Pool and Non-Constant Pool allows duplicates.**

**-> String class has automatically overridden 3 methods from Object class**

**1. toString()**

**2. hashCode()**

**3. equals()**

**-> toString() of the Object Class is Overridden in String Class to return the actual data passed to the Constructor during object creation.**

**-> hashCode() of the Object Class is overridden in String class to return a number based on the ASCII value.**

**-> equals() of the Object Class is Overridden in String Class to compare the contents of both the objects.**

**1a.**

**package** jspiders;

**public** **class** Car {

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

Car c = **new** Car();

System.***out***.println(c);

System.***out***.println(c.toString());

System.***out***.println("------------");

System.***out***.println(c.hashCode());

System.***out***.println("------------");

Car c1 = **new** Car();

Car c2 = **new** Car();

System.***out***.println(c1.equals(c2)); // false

System.***out***.println(c1 == c2); // false

}

}

o/p:

jspiders.Car@15db9742

jspiders.Car@15db9742

------------

366712642

------------

false

false

1b.

**package** jspiders;

**public** **class** Demo {

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

String s = **new** String("A");

System.***out***.println(s);

System.***out***.println(s.toString());

System.***out***.println("---------");

System.***out***.println(s.hashCode());

System.***out***.println("---------");

String s1 = **new** String("java");

String s2 = **new** String("java");

System.***out***.println(s1.equals(s2)); // true

System.***out***.println(s1 == s2); // false

}

}

o/p:

A

A

---------

65

---------

true

false

**Constructors in String Class**

**package jspiders;**

**public class Test {**

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

**// Empty Representation of a String Object**

**String s1 = new String();**

**System.out.println(s1);**

**// Passing a String Object**

**String s2 = new String("dinga");**

**System.out.println(s2);**

**char[] ch = {'j', 'a', 'v', 'a'};**

**// Converting an Array of Characters to String**

**String s3 = new String(ch);**

**System.out.println(s3);**

**}**

**}**

**o/p:**

**dinga**

**java**

**1. How String Objects are Immutable? Explain String Immutability Concept.**

**- When we re-initialize a String object, rather than modifying the same object, a new object is created and the reference pointing to the old object gets de-referenced and starts pointing to the newly created object.**

**This is String Immutability Concept.**

**Mutable Version of String**

**1. StringBuffer**

**2. StringBuilder**

**--------------------------------------**

**1.**

**package jspiders;**

**public class Student {**

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

**Student s = new Student();**

**System.out.println(s); // implicitly calls toString()**

**System.out.println(s.toString()); // explicitly calling toString()**

**System.out.println("-----------------");**

**Student std = new Student();**

**System.out.println(std.hashCode());**

**System.out.println("-----------------");**

**Student s1 = new Student();**

**Student s2 = new Student();**

**System.out.println(s1.equals(s2));**

**}**

**}**

**o/p:**

**jspiders.Student@15db9742**

**jspiders.Student@15db9742**

**-----------------**

**1829164700**

**-----------------**

**false**

**2.**

**package jspiders;**

**public class Demo {**

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

**// String s = "java";**

**String s = new String("java");**

**System.out.println(s); // implicitly calls toString()**

**System.out.println(s.toString()); // explicitly calling toString()**

**System.out.println("-----------------");**

**String s1 = new String("a");**

**System.out.println(s1.hashCode());**

**System.out.println("-----------------");**

**String a = new String("Dinga");**

**String b = new String("Dinga");**

**System.out.println(a.equals(b));**

**}**

**}**

**o/p:**

**java**

**java**

**-----------------**

**97**

**-----------------**

**true**

**Methods Present in String class**

**package** jspiders;

**public** **class** MethodsDemo {

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

String s = "Software Developer";

System.***out***.println(s.length()); // 18

System.***out***.println("------------------");

System.***out***.println(s.toUpperCase());

System.***out***.println("------------------");

System.***out***.println(s.toLowerCase());

System.***out***.println("------------------");

System.***out***.println(s.startsWith("soft"));

System.***out***.println(s.startsWith("Soft"));

System.***out***.println("------------------");

System.***out***.println(s.endsWith("er"));

System.***out***.println(s.endsWith("Eloper"));

System.***out***.println("------------------");

System.***out***.println(s.contains("dev"));

System.***out***.println(s.contains("Dev"));

System.***out***.println("------------------");

System.***out***.println(s.concat(" in TY"));

System.***out***.println("------------------");

// Software Developer

System.***out***.println(s.charAt(2));

System.***out***.println(s.charAt(14));

System.***out***.println("------------------");

System.***out***.println(s.indexOf('t'));

System.***out***.println(s.indexOf('D'));

System.***out***.println(s.indexOf('e'));

System.***out***.println("------------------");

String a = "java";

String b = "JavA";

String c = "java";

System.***out***.println(a.equals(b)); // false

System.***out***.println(a.equals(c)); // true

System.***out***.println(a.equalsIgnoreCase(b));

System.***out***.println("------------------");

String x = "hello dinga";

System.***out***.println(x.substring(3)); // lo dinga

System.***out***.println(x.substring(7)); // inga

System.***out***.println(x.substring(2, 7)); // llo d

System.***out***.println(x.substring(4, 10)); // o ding

System.***out***.println("------------------");

}

}

o/p:

18

------------------

SOFTWARE DEVELOPER

------------------

software developer

------------------

false

true

------------------

true

false

------------------

false

true

------------------

Software Developer in TY

------------------

f

o

------------------

3

9

7

------------------

false

true

true

------------------

lo dinga

inga

llo d

o ding

1.

**package** jspiders;

**public** **class** Test {

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

String s = "Hello World";

System.***out***.println(s.substring(3));

System.***out***.println(s.substring(6));

System.***out***.println(s.substring(2, 8));

System.***out***.println(s.substring(1,7));

System.***out***.println("-----------------");

String a = "java";

**char**[] ch = a.toCharArray();

**for**(**int** i=ch.length-1; i>=0; i--) {

System.***out***.println(ch[i]);

}

System.***out***.println("-----------------");

String b = "dinga";

**for**(**int** i=b.length()-1; i>=0; i--){

System.***out***.println(b.charAt(i));

}

}

}

// Palindrome -> Assignment

o/p:

lo World

World

llo Wo

ello W

-----------------

a

v

a

j

-----------------

a

g

n

i

d