**Strings**

**1. Different ways creating a string**

String str="Hello!";

String str1= new String("Hello!");

char ch[]={ 'H','e','l','l','o','!',};

String str1=new String(ch);

**2. Concatenating two strings using + operator**

class TestStringConcatenation1{

public static void main(String args[]){

String s="Sachin"+" Tendulkar";

System.out.println(s);//Sachin Tendulkar

}

}

**3. Finding the length of the string**

public class LengthExample{

public static void main(String args[]){

String s1="javatpoint";

String s2="python";

System.out.println("string length is: "+s1.length());

System.out.println("string length is: "+s2.length());

}}

**4. Extract a string using Substring**

public String substring(int beginIndex) {

if (beginIndex < 0) {

throw new StringIndexOutOfBoundsException(beginIndex);

}

int subLen = value.length - beginIndex;

if (subLen < 0) {

throw new StringIndexOutOfBoundsException(subLen);

}

return (beginIndex == 0) ? this : new String(value, beginIndex, subLen);

}

**5. Searching in strings using indexOf()**

public class IndexOfExample{

public static void main(String args[]){

String s1="this is index of example";

int index1=s1.indexOf("is");

int index2=s1.indexOf("index");

System.out.println(index1+" "+index2);

int index3=s1.indexOf("is",4);

System.out.println(index3);

int index4=s1.indexOf('s');

System.out.println(index4);

}}

**6. Matching a String Against a Regular Expression With matches()**

import java.io.\*;

public class Test {

public static void main(String args[]) {

String Str = new String("Welcome to Tutorialspoint.com");

System.out.print("Return Value :" );

System.out.println(Str.matches("(.\*)Tutorials(.\*)"));

System.out.print("Return Value :" );

System.out.println(Str.matches("Tutorials"));

System.out.print("Return Value :" );

System.out.println(Str.matches("Welcome(.\*)"));

}

}

**7. Comparing strings using the methods equals()**

public class GFG {

public static int stringCompare(String str1, String str2)

{

int l1 = str1.length();

int l2 = str2.length();

int lmin = Math.min(l1, l2);

for (int i = 0; i < lmin; i++) {

int str1\_ch = (int)str1.charAt(i);

int str2\_ch = (int)str2.charAt(i);

if (str1\_ch != str2\_ch) {

return str1\_ch - str2\_ch;

}

}

if (l1 != l2) {

return l1 - l2;

}

else {

return 0;

}

}

public static void main(String args[])

{

String string1 = new String("Hello");

String string2 = new String("world");

String string3 = new String("hie");

String string4 = new String("hey");

System.out.println("Comparing " + string1 + " and " + string2

+ " : " + stringCompare(string1, string2));

System.out.println("Comparing " + string3 + " and " + string4

+ " : " + stringCompare(string3, string4));

System.out.println("Comparing " + string1 + " and " + string4

+ " : " + stringCompare(string1, string4));

}

}

**8. equalsIgnoreCase(), startsWith(), endsWith() and compareTo()**

public class StringComparison {

public static void main(String[] args) {

String str = "This is a test string";

String str1 = new String("This");

System.out.println("String for comparison -- " + str.substring(0, 4));

System.out.println(str.substring(0, 4).equals("This"));

System.out.println(str.substring(0, 4).equals("this"));

System.out.println(str.substring(0, 4).equalsIgnoreCase("this"));

System.out.println(str1.equalsIgnoreCase("this"));

}

}

**9. Trimming strings with trim()**

public String trim() {

int len = value.length;

int st = 0;

char[] val = value; /\* avoid getfield opcode \*/

while ((st < len) && (val[st] <= ' ')) {

st++;

}

while ((st < len) && (val[len - 1] <= ' ')) {

len--;

}

return ((st > 0) || (len < value.length)) ? substring(st, len) : this;

}

**10. Replacing characters in strings with replace()**

public String replace(char oldChar, char newChar) {

if (oldChar != newChar) {

int len = value.length;

int i = -1;

char[] val = value; /\* avoid getfield opcode \*/

while (++i < len) {

if (val[i] == oldChar) {

break;

}

}

if (i < len) {

char buf[] = new char[len];

for (int j = 0; j < i; j++) {

buf[j] = val[j];

}

while (i < len) {

char c = val[i];

buf[i] = (c == oldChar) ? newChar : c;

i++;

}

return new String(buf, true);

}

}

return this;

}

**11. Splitting strings with split()**

public class GFG {

public static void main(String args[])

{

String str = "geekss@for@geekss";

String[] arrOfStr = str.split("@", 2);

for (String a : arrOfStr)

System.out.println(a);

}

}

**12. Converting Numbers to Strings with valueOf()**

public class IntToStringExample1{

public static void main(String args[]){

int i=200;

String s=String.valueOf(i);

System.out.println(i+100);

System.out.println(s+100);

}}

**13. Converting integer objects to Strings**

class GFG {

public static void main(String args[])

{

int a = 1234;

int b = -1234;

String str1 = Integer.toString(a);

String str2 = Integer.toString(b);

System.out.println("String str1 = " + str1);

System.out.println("String str2 = " + str2);

}

}

**14. Converting to uppercase and lowercase**

String txt = "Hello World";

System.out.println(txt.toUpperCase());

System.out.println(txt.toLowerCase());