Important methods of string class:

1)public char charAt(int index);

Returns the character locating at specified index.

2)public String concat(String str);

3)public Boolean equals(Object obj);

Used for case sensitive comparision

4)public Boolean equalsIgnoreCase(Strings);

For content comparison where case is not important.

5)public String substring(int begin);

Return the substring from begin index to end of the string.

6)public String substring(int begin, int end);

Returns the substring from begin index to end-1 index.

7)public int length();

Returns the number of characters present in the string.

8)public String replace(char old, char new);

To replace every old character with a new character.

9)public String to LowerCase();

Converts the all characters of the string to lowercase.

10)public String to UpperCase();

Converts the all characters of the string to uppercase.

11)public String trim();

We can use this method to remove blank spaces present at beginning and end of

The string but not blank spaces present at middle of the String.

12)public int indexOf(char ch);

It returns index of 1st occurrence of the specified character if the specified

Character is not available then return-1.

13)public int lastIndexOf(Char ch);

It returns index of last occurrence of the specified character if the specified

Character is not available then return -1

1. Design the program to display velocity message on screen

**public** **class** SampleTest {

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

String str = "velocity";

System.***out***.println("Institute name is>>" + str);

}

}

1. Design the method to return the name

**public** **class** StringDemo {

**private** **static** String getStudentName(String name) {

**return** name;

}

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

System.***out***.println("Enter your name>>");

Scanner scanner= **new** Scanner(System.***in***);

String name=scanner.next();

String s= *getStudentName*(name);

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

}

}

1. Design the program to perform the string operation

**public** **class** SampleTest {

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

String str = "velocity";

System.***out***.println(str.length());

System.***out***.println(str.charAt(4));

System.***out***.println(str.compareTo("velocity"));

System.***out***.println(str.concat("pune"));

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

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

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

}

}

1. Design the program to counting space into string.

**public** **class** SampleTest {

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

String str = "velocity training center pune";

**int** counter = 0;

**for** (**int** i = 0; i < str.length(); i++) {

**char** ch = str.charAt(i);

**if** (ch == ' ') {

counter++;

}

}

System.***out***.println("total space in string are>>" + counter);

}

}

**String Buffer**

It provides us with a way to use mutable strings in java. These strings are safe to be used by multiple threads simultaneously. In order to give this advantage to the StringBuffer, the implementation of this class becomes less time efficient.

StringBuffer a= new StringBuffer("Hello");

System.out.println(a);

//updating the string

a.append(" World");

System.out.println(a);

**String Builder**

StringBuilder also provides us with mutable strings but here we lack thread safety. It cannot be used by multiple threads simultaneously. Since the class is not applying this extra feature like StringBuffer, it becomes faster than it.

StringBuilder a= new StringBuilder("String");

System.out.println(a);

//updating the string

a.append(" Articles");

System.out.println(a);

DifferencebetweenString,StringBufferandStringBuilder(SelfAssignment)