**Strings (Java) :**

### 1.String : CompareTo()

Use the method “compareTo” and specify the String that you would like to compare.

Use “compareToIgnoreCase” in case you don’t want the result to be case sensitive.

The result will have the value 0 if the argument string is equal to this string; a value less than 0 if this string is lexicographically less than the string argument; and a value greater than 0 if this string is lexicographically greater than the string argument.

### 2. String Contain() Method

I partially know what the string should have contained, how do I confirm if the String contains a sequence of characters I specify?

Use the method “contains” and specify the characters you need to check.

Returns **true** if and only if this string contains the specified sequence of char values.

### 3. String endsWith() Method

How do I confirm if a String ends with a particular suffix? Again you answered it. Use the “endsWith” method and specify the suffix in the arguments.

Returns **true** if the character sequence represented by the argument is a suffix of the character sequence represented by this object.

### 4.String replaceAll() & replaceFirst()

I want to modify my String at several places and replace several parts of the String?

Java String Replace, replaceAll() and replaceFirst() methods. You can specify the part of the String you want to replace and the replacement String in the arguments.

## **The String Class in C++**

The standard C++ library provides a **string** class type that supports all the operations mentioned above, additionally much more functionality. Let us check the following example −

#include <iostream>

#include <string>

using namespace std;

int main () {

string str1 = "Hello";

string str2 = "World";

string str3;

int len ;

// copy str1 into str3

str3 = str1;

cout << "str3 : " << str3 << endl;

// concatenates str1 and str2

str3 = str1 + str2;

cout << "str1 + str2 : " << str3 << endl;

// total length of str3 after concatenation

len = str3.size();

cout << "str3.size() : " << len << endl;

return 0;

}

When the above code is compiled and executed, it produces result something as follows −

str3 : Hello

str1 + str2 : HelloWorld

str3.size() : 10