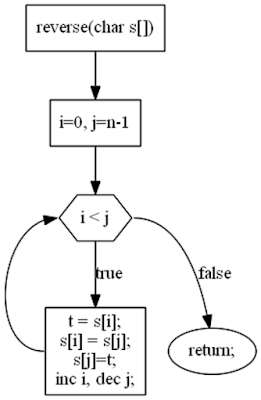
**How to Reverse a String in place in Java - Example**

One of the common Java coding interview questions is to *write a program to reverse a String in place in Java, without using additional memory*. You cannot use any library classes or methods like e.g. [StringBuilder](http://www.java67.com/2014/05/difference-between-stringbuilder-and-StringBuffer-java.html) to solve this problem. This restriction is placed because StringBuilder and StringBuffer class define a reverse() method which can easily reverse the given String. Since the main objective of this question is to test the programming skill and coding logic of candidate, there is no point giving him the option to use the library method which can make this question trivial. Now, how do you solve this problem? If you are familiar with array data structure then it would be easy for you. Since String is backed by a character array, you can use the same in place algorithm we have used to [reverse an array in place](http://java67.blogspot.com/2016/01/java-program-to-reverse-array-in-place.html).  
  
That technique uses the two-pointer approach where one pointer starts from the beginning and other pointer starts from the end of the array. You swap elements until they meet. At that point in time, your String or array is already reversed.  
  
This is an acceptable solution because we have not used additional memory and any library method, but you can also be asked to explain about the time and space complexity of your solution.  
  
The time complexity of this algorithm is O(n/2) + time taken in swapping, which effectively adds up to **O(n)** time. This means time will increase in the proportion of the length of String or number of characters on it.  The space complexity is O(1) because we are not using any additional memory to reverse the String.  
  
Btw, String is a very popular topic on interviews and you will often see a couple of String based coding questions on interviews. A good knowledge of String along with other data structures like an array, linked list, and binary tree is very important. If you feel you lack that knowledge or want to improve it, I suggest you take a look at [Data Structures and Algorithms: Deep Dive Using Java](https://click.linksynergy.com/fs-bin/click?id=JVFxdTr9V80&subid=0&offerid=323058.1&type=10&tmpid=14538&RD_PARM1=https%3A%2F%2Fwww.udemy.com%2Fdata-structures-and-algorithms-deep-dive-using-java%2F) course on Udemy. It's both affordable and very comprehensive course and I highly recommend it for Java programmers.

**Java Program to Reverse a String in place**

Here is the simple example to reverse characters in String by using two pointer technique. This is an **in-place algorithm** because it doesn't allocate any extra array, it just uses the two int variables to hold positions from start and end.  
  
If you look closely this algorithm is similar to the algorithm we have earlier used to [reverse an array in place](http://java67.blogspot.com/2016/01/java-program-to-reverse-array-in-place.html). That's obvious because String is backed by character array in Java.  
  
If you know how to reverse an array in place then reversing a String is not different for you.  What is more important is checking for [null and empty String](http://javarevisited.blogspot.com/2016/01/how-to-check-if-string-is-not-null-and-empty-in-java-example.html) because this is where many programmers become lazy and started writing code without validating input.  
  
You must write your best code during programming interviews. The code which can stand the test of time in production is what every interview like to see.  If you don't know how to write production quality code, I suggest you take a look at [Clean Code](http://www.amazon.com/Clean-Code-Handbook-Software-Craftsmanship/dp/0132350882?tag=javamysqlanta-20), one of the books you should read at the start of your programming career.  
  
Btw, if you are not familiar with recursion and iteration or basic fundamentals of Data Structures and Algorithms then I suggest you join a comprehensive course like [Algorithms and Data Structures - Part 1 and 2](https://pluralsight.pxf.io/c/1193463/424552/7490?u=https%3A%2F%2Fwww.pluralsight.com%2Fcourses%2Fads-part1) on Pluralsight to learn them well. It makes a lot of sense to solve problems like this when you know fundamentals better.  
  
Here is the iterative algorithm to reverse String in place:

[](https://pluralsight.pxf.io/c/1193463/424552/7490?u=https://www.pluralsight.com/courses/ads-part1)

Btw, you would need a Pluralsight membership to access this course, which costs around $29 monthly or $299 annually. I have one and I also suggest all developers have that plan because Pluralsight is like NetFlix for Software developers.  
  
It has more than 5000+ good quality courses on all latest topics. Since we programmers have to learn new things every day, an investment of $299 USD is not bad.  
  
Btw, it also offers a [**10-day free trial**](http://pluralsight.pxf.io/c/1193463/424552/7490?u=https%3A%2F%2Fwww.pluralsight.com%2Flearn) without any obligation which allows you to watch 200 hours of content. You can watch this course for free by signing for that trial.

**Java Program to reverse a String in Place**

import org.junit.Assert;

import org.junit.Test;

/\*\*

\* Java Program to [reverse a String in place](https://javarevisited.blogspot.com/2016/03/how-to-reverse-string-in-place-in-java.html),

\* without any additional buffer in Java.

\*

\* @author WINDOWS 8

\*

\*/

public class StringReversal {

/\*\*

\* Java method to reverse a String in place

\* @param str

\* @return reverse of String

\*/

public static String reverse(String str) {

if(str == null || str.isEmpty()){

return str;

}

char[] characters = str.toCharArray();

int i = 0;

int j = characters.length - 1;

while (i < j) {

swap(characters, i, j);

i++;

j--;

}

return new String(characters);

}

/\*\*

\* Java method to swap two numbers in given array

\* @param str

\* @param i

\* @param j

\*/

private static void swap(char[] str, int i, int j) {

char temp = str[i];

str[i] = str[j];

str[j] = temp;

}

@Test

public void reverseEmptyString(){

Assert.assertEquals("", reverse(""));

}

@Test

public void reverseString(){

Assert.assertEquals("cba", reverse("abc"));

}

@Test

public void reverseNullString(){

Assert.assertEquals(null, reverse(null));

}

@Test

public void reversePalindromeString(){

Assert.assertEquals("aba", reverse("aba"));

}

@Test

public void reverseSameCharacterString(){

Assert.assertEquals("aaa", reverse("aaa"));

}

@Test

public void reverseAnagramString(){

Assert.assertEquals("mary", reverse("yram"));

}

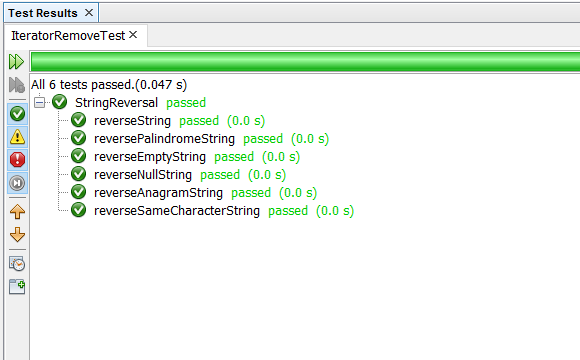
}

You might have also noticed that this time, I have not to use the [main() method](http://java67.blogspot.com/2015/06/can-you-overload-or-override-main-in-java.html) to test the code, instead I have written couple of JUnit test cases. It's actually better to write unit test cases all the time to test your code instead of using main() method as it put unit testing in your habit.  
  
Btw, if you feel reluctance on writing unit tests or not sure how to write tests, I suggest you read [**Test Driven**](http://www.amazon.com/Test-Driven-Acceptance-Java-Developers/dp/1932394850?tag=javamysqlanta-20), one of the best book on Test drive development but even if you don't follow TDD, it will help you to write better code and unit tests.

I have written following JUnit tests to check whether our reverse method is working for different kinds of String or not, the JUnit test result is also attached below:

* Unit test to reverse an empty String
* Test to reverse a null String
* Reverse a [palindrome](http://javarevisited.blogspot.com/2015/06/2-ways-to-check-if-string-is-palindrome-java.html) String
* Unit tests to reverse a one character string
* JUnit test to reverse a string with the same character
* Reverse a String with a different character
* Reverse an [anagram](http://javarevisited.blogspot.sg/2013/03/Anagram-how-to-check-if-two-string-are-anagrams-example-tutorial.html) String

And, here is the result of running these JUnit tests:

[](https://click.linksynergy.com/fs-bin/click?id=JVFxdTr9V80&subid=0&offerid=323058.1&type=10&tmpid=14538&RD_PARM1=https://www.udemy.com/data-structures-and-algorithms-deep-dive-using-java/)

You can see that all the unit tests are passing, which is good. You can also add more unit test to further test our method of reversing String in Java.  
  
That's all about **how to reverse String in place in Java**. This is a common algorithm which uses two pointer approach. Since it requires us to traverse the array till middle, time complexity is O(n/2) i.e. O(n). It doesn't use any external buffer instead just use two variables to keep track of indices from start and end.  
  
**Further Learning**  
[Data Structures and Algorithms: Deep Dive Using Java](https://click.linksynergy.com/fs-bin/click?id=JVFxdTr9V80&subid=0&offerid=323058.1&type=10&tmpid=14538&RD_PARM1=https%3A%2F%2Fwww.udemy.com%2Fdata-structures-and-algorithms-deep-dive-using-java%2F)  
[Algorithms and Data Structures - Part 1 and 2](https://pluralsight.pxf.io/c/1193463/424552/7490?u=https%3A%2F%2Fwww.pluralsight.com%2Fcourses%2Fads-part1)  
[Cracking the Coding Interview - 189 Questions and Solutions](http://www.amazon.com/Cracking-Coding-Interview-6th-Edition/dp/0984782850/?tag=javamysqlanta-20)

public class **StringReverseExample** {

    public static void **main**(String args[]) throws FileNotFoundException, IOException {

        //original string

        String str = "Sony is going to introduce Internet TV soon";

        System.out.println("Original String: " + str);

**//reversed string using Stringbuffer**

        String reverseStr = new StringBuffer(str).reverse().toString();

        System.out.println("Reverse String in Java using StringBuffer: " + reverseStr);

**//iterative method to reverse String in Java**

        reverseStr = reverse(str);

        System.out.println("Reverse String in Java using Iteration: " + reverseStr);

**//recursive method to reverse String in Java**

        reverseStr = reverseRecursively(str);

        System.out.println("Reverse String in Java using Recursion: " + reverseStr);

    }

    public static String **reverse**(String str) {

        StringBuilder strBuilder = new StringBuilder();

        char[] strChars = str.toCharArray();

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

            strBuilder.append(strChars[i]);

        }

        return strBuilder.toString();

    }

    public static String **reverseRecursively**(String str) {

**//base case to handle one char string and empty string**

        if (str.length() < 2) {

            return str;

        }

        return reverseRecursively(str.substring(1)) + str.charAt(0);

    }

}