static – StringManip FRQ

Assume that the classes listed in the Java Quick Reference have been imported where appropriate.
Unless otherwise noted in the question, assume that parameters in method calls are not null and that methods are called only when their preconditions are satisfied.
In writing solutions for each question, you may use any of the accessible methods that are listed in classes defined in that question. Writing significant amounts of code that can be replaced by a call to one of these methods will not receive full credit.

Turn (a), (b) & (c) into static methods called in a main method either in the same class or separate class (you choose).

This question involves the *StringManip* class, which is used to perform manipulation on strings.

(a) Write a code segment, which takes the String variable str and prints a new string with spaces removed. For example, if str points to *"hi how are you"*, the code segment should print *"hihowareyou"*.

Complete the code segment below.

```
/** Takes a string str and prints a new string
 *  with all spaces removed.
 */
String str = "hi how are you";
```

(b) A proceeding code segment in the *StringManip* class, which takes the String produced by part *(a)* *(str with spaces removed)* and prints a new string with the characters in reverse order. For example, if str with spaces removed is *"ABCDE"* the code segment should print *"EDCBA"*.

Complete the code segment below by assigning the reversed string to reverseString.

```
/** Takes a string str and returns a new string
 *  with the characters reversed.
 */
String reverseString = "";
System.out.println(reverseString);
```

(c) For this question, let a palindrome be defined as a string that, when spaces are removed, reads the same forward and backward. For example, `"race car"` and `"taco cat"` are palindromes. You will write a code segment, which determines whether the `String str` is a palindrome and prints a message indicating the result. Examples of the intended behavior of the method are shown in the following table.

`String str`	Printed Message
`"taco cat"`	`taco cat is a palindrome`
`"laid on no dial"`	`laid on no dial is a palindrome`
`"level up"`	`level up is not a palindrome`

Write code segment below. Assume that previous code segments in parts *(a)* & *(b)* works as specified, regardless of what you wrote in them. You must use results of the previous code segments in parts *(a)* & *(b)* appropriately to receive full credit. Your implementation must conform to the examples in the table.

```
/** Determines whether str is a palindrome and prints a
 *  message indicating the result, as described in part (c).
 *  Precondition: str contains only lowercase letters and
 *  spaces.
*/
```