Worksheet: String Class Coding Practice

1. Write a `static` method named `middleCharacters` that takes in a `String` parameter and returns a `String` that is:
   - the **middle character**, if the word has an odd number of characters;
   - the **middle two characters** if the word has an even number of characters.

   You do not need to test if the input `String` parameter is `null`; however, for the more advanced programmer, account for the case where the input `String` parameter is an empty string.

   The following is example test code:

```
System.out.println(middleCharacters("12345"));
System.out.println(middleCharacters("abcdef"));
System.out.println(middleCharacters("x"));
System.out.println(middleCharacters("xy"));
System.out.println(middleCharacters("Hello World!"));
//System.out.println(middleCharacters("")); // Advanced
```

   And the output expected from the above test code is:

```
3
cd
x
xy
 W
```

   Write your completed code in the space below:

Worksheet: String Class Coding Practice

2. Write a `static` method named `removeFirstWord` that takes in a `String` parameter and returns a `String` that is the same as the input string, except has the first word removed. For this exercise, you can assume the first and second word are separated by a single space character. Use only the *AP Java Subset* `String` class methods and no other library methods. You must take into account the possibility of the input `String` only containing a single word, or being an empty string. You do not need to check for a `null` input. The following is example test code:

```
System.out.println(removeFirstWord("Hello World!"));
System.out.println(removeFirstWord("Only two words."));
System.out.println(removeFirstWord("Emptystring"));
System.out.println(removeFirstWord(" Two words."));
```

And the output expected from the above test code is:

```
World!
two words.

Two words.
```

Write your completed code in the space below: