Check if First String Contains Second String:

```
import java.util.Scanner;

public class StringContains {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the first string:");
        String str1 = scanner.nextLine();
        System.out.println("Enter the second string:");
        String str2 = scanner.nextLine();
        scanner.close();

        if (str1.contains(str2)) {
            System.out.println("First string contains the second string.");
        } else {
            System.out.println("First string does not contain the second
string.");
        }
    }
}
```

Output

Enter the first string:
bibek sah
Enter the second string:
bibek
First string contains the second string.

Swap Two Strings Without Using Third Variable:

```
public class StringSwap {
    public static void main(String[] args) {
        String str1 = "Hello";
        String str2 = "World";

        System.out.println("Before swapping: str1 = " + str1 + ", str2 = " +
str2);

        str1 = str1 + str2;
        str2 = str1.substring(0, str1.length() - str2.length());
        str1 = str1.substring(str2.length());

        System.out.println("After swapping: str1 = " + str1 + ", str2 = " +
str2);
    }
}
```

Output

Before swapping: str1 = Hello, str2 = World
After swapping: str1 = World, str2 = Hello

Find First Non-Repeated Character in a String:

```
public class FirstNonRepeatedChar {
    public static void main(String[] args) {
        String str = "hello";

        for (int i = 0; i < str.length(); i++) {
            char ch = str.charAt(i);
            if (str.indexOf(ch) == str.lastIndexOf(ch)) {
                System.out.println("First non-repeated character: " + ch);
                break;
            }
        }
    }
}
```

Output
First non-repeated character: h

Check if a String Contains Only Digits:
i) Using Character.isDigit():

```
public class CheckDigits {
    public static void main(String[] args) {
        String str = "12345";
        boolean containsOnlyDigits = str.chars().allMatch(Character::isDigit);
        System.out.println("Contains only digits: " + containsOnlyDigits);
    }
}
```

output
Contains only digits: true

ii) Using Regular Expression:

```
public class CheckDigitss {
    public static void main(String[] args) {
        String str = "12345";
        boolean containsOnlyDigits = str.matches("[0-9]+");
        System.out.println("Contains only digits: " + containsOnlyDigits);
    }
}
```

Output
Contains only digits: true

Remove All Occurrences of a Given Character from a String:

```
public class RemoveCharacter {
    public static void main(String[] args) {
        String str = "Hello";
        char ch = 'l';
        String modifiedStr = str.replaceAll(String.valueOf(ch), "");
        System.out.println("Modified string: " + modifiedStr);
    }
}
```

Output

Modified string: Heo

Check if a String is Palindrome:

```
public class Palindrome {
    public static void main(String[] args) {
        String str = "madam";
        boolean isPalindrome = true;
        for (int i = 0; i < str.length() / 2; i++) {
            if (str.charAt(i) != str.charAt(str.length() - i - 1)) {
                isPalindrome = false;
                break;
            }
        }
        if (isPalindrome) {
            System.out.println("String is palindrome.");
        } else {
            System.out.println("String is not palindrome.");
        }
    }
}
```

Output

String is palindrome.

Convert an Array to String in Java:

```
import java.util.Arrays;

public class ArrayToString {
    public static void main(String[] args) {
        int[] arr = {1, 2, 3, 4, 5};
        String str = Arrays.toString(arr);
        System.out.println("Array as string: " + str);
    }
}
```

Output

Array as string: [1, 2, 3, 4, 5]

Print Duplicate Characters from a String:

```
import java.util.HashMap;
import java.util.Map;

public class DuplicateChars {
    public static void main(String[] args) {
        String str = "hello";
        Map<Character, Integer> charCountMap = new HashMap<>();

        for (char ch : str.toCharArray()) {
            charCountMap.put(ch, charCountMap.getOrDefault(ch, 0) + 1);
        }

        System.out.println("Duplicate characters:");
        for (Map.Entry<Character, Integer> entry : charCountMap.entrySet()) {
            if (entry.getValue() > 1) {
                System.out.println(entry.getKey() + ": " + entry.getValue() +
" times");
            }
        }
    }
}
```

Output

Duplicate characters:

l: 2 times

Check if Two Strings are Anagrams:

```
import java.util.Arrays;

public class AnagramCheck {
    public static void main(String[] args) {
        String str1 = "listen";
        String str2 = "silent";
        boolean areAnagrams = true;

        if (str1.length() != str2.length()) {
            areAnagrams = false;
        } else {
            char[] charArray1 = str1.toCharArray();
            char[] charArray2 = str2.toCharArray();
            Arrays.sort(charArray1);
            Arrays.sort(charArray2);
```

```
            areAnagrams = Arrays.equals(charArray1, charArray2);
        }

        if (areAnagrams) {
            System.out.println("Strings are anagrams.");
        } else {
            System.out.println("Strings are not anagrams.");
        }
    }
}
```

Output

Strings are anagrams.

Reverse a String in Java without Using the Reverse Method:

```
public class ReverseString {
    public static void main(String[] args) {
        String str = "hello";
        StringBuilder reversedStr = new StringBuilder();

        for (int i = str.length() - 1; i >= 0; i--) {
            reversedStr.append(str.charAt(i));
        }

        System.out.println("Reversed string: " + reversedStr);
    }
}
```

Output

Reversed string: olleh