String Programming Assignment - Java Solutions

```java
import java.util.*;

public class StringPrograms {

    public static String removeDuplicates(String str) {
        StringBuilder sb = new StringBuilder();
        HashSet<Character> seen = new HashSet<>();
        for (char ch : str.toCharArray()) {
            if (!seen.contains(ch)) {
                sb.append(ch);
                seen.add(ch);
            }
        }
        return sb.toString();
    }

    public static List<Character> printDuplicates(String str) {
        Map<Character, Integer> countMap = new HashMap<>();
        List<Character> duplicates = new ArrayList<>();
        for (char ch : str.toCharArray()) {
            countMap.put(ch, countMap.getOrDefault(ch, 0) + 1);
        }
        for (Map.Entry<Character, Integer> entry : countMap.entrySet()) {
            if (entry.getValue() > 1) {
                duplicates.add(entry.getKey());
            }
        }
        return duplicates;
    }

    public static boolean isPalindrome(String str) {
        return str.equals(new StringBuilder(str).reverse().toString());
    }

    public static Map<String, Integer> countCharacters(String str) {
        int vowels = 0, consonants = 0, specialChars = 0;
        String vowelSet = "AEIOUaeiou";
        for (char ch : str.toCharArray()) {
            if (Character.isLetter(ch)) {
                if (vowelSet.indexOf(ch) != -1) {
                    vowels++;
                } else {
                    consonants++;
                }
            } else if (!Character.isWhitespace(ch)) {
                specialChars++;
            }
        }
        Map<String, Integer> result = new HashMap<>();
        result.put("vowels", vowels);
        result.put("consonants", consonants);
        result.put("special_chars", specialChars);
        return result;
    }

    public static boolean isAnagram(String str1, String str2) {
        char[] arr1 = str1.replaceAll("\s", "").toLowerCase().toCharArray();
        char[] arr2 = str2.replaceAll("\s", "").toLowerCase().toCharArray();
        Arrays.sort(arr1);
        Arrays.sort(arr2);
        return Arrays.equals(arr1, arr2);
    }

    public static boolean isPangram(String str) {
        str = str.toLowerCase();
        for (char ch = 'a'; ch <= 'z'; ch++) {
            if (!str.contains(String.valueOf(ch))) {
                return false;
            }
        }
        return true;
    }

    public static boolean hasUniqueChars(String str) {
        Set<Character> charSet = new HashSet<>();
        for (char ch : str.toCharArray()) {
            if (!charSet.add(ch)) {
                return false;
            }
        }
        return true;
    }

    public static char maxOccurringChar(String str) {
        Map<Character, Integer> countMap = new HashMap<>();
        for (char ch : str.toCharArray()) {
            countMap.put(ch, countMap.getOrDefault(ch, 0) + 1);
        }
                                          return     Collections.max(countMap.entrySet(),
Map.Entry.comparingByValue()).getKey();
    }

    public static void main(String[] args) {
        System.out.println("Remove Duplicates: " + removeDuplicates("programming"));
        System.out.println("Duplicate Characters: " + printDuplicates("hello world"));
        System.out.println("Is '2552' Palindrome: " + isPalindrome("2552"));
        System.out.println("Character Counts: " + countCharacters("Hello, World!"));
         System.out.println("Are 'listen' and 'silent' Anagrams: " + isAnagram("listen",
"silent"));
          System.out.println("Is Pangram: " + isPangram("The quick brown fox jumps over
the lazy dog"));
        System.out.println("Has Unique Characters: " + hasUniqueChars("abcdefg"));
              System.out.println("Max  Occurring  Character:  "  +  maxOccurringChar("hello
world"));
```

```
    }
}
```