1. Reverse a String

Write a program to reverse a given string without using the built-in reverse() method.

Example:

- Input: "Java"
- Output: "avaJ"

java Copy Edit

```java
public class ReverseString {
    public static void main(String[] args) {
        String str = "Java";
        String reversed = "";
        for (int i = str.length() - 1; i >= 0; i--) {
            reversed += str.charAt(i);
        }
        System.out.println("Reversed String: " + reversed);
    }
}
```

2. Check if a String is Palindrome

Write a program to check if a given string is a palindrome (reads the same backward and forward).

Example:

- Input: `"madam"`
- Output: `true`

java Copy Edit

```java
public class Palindrome {
    public static void main(String[] args) {
        String str = "madam";
        String reversed = new StringBuilder(str).reverse().toString();
        if (str.equals(reversed)) {
            System.out.println("The string is a palindrome.");
        } else {
            System.out.println("The string is not a palindrome.");
        }
    }
}
```

3. Count the Number of Vowels in a String

Write a program to count the number of vowels in a given string.

Example:

- Input: `"Automation"`
- Output: `5`

```java
public class CountVowels {
    public static void main(String[] args) {
        String str = "Automation";
        int count = 0;
        for (int i = 0; i < str.length(); i++) {
            char ch = str.charAt(i);
            if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||
                ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U') {
                count++;
            }
        }
        System.out.println("Number of vowels: " + count);
    }
}
```

4. Count Occurrences of a Character in a String

Write a program to count the occurrences of a specific character in a string.

Example:

- Input: `"hello world"`, Character: `'o'`
- Output: `2`

Java

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

public class CharacterCounter {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String inputString = scanner.nextLine();
        scanner.close();

        Map<Character, Integer> charCountMap = new HashMap<>();

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

        System.out.println("Character occurrences:");
        for (Map.Entry<Character, Integer> entry : charCountMap.entrySet()) {
            System.out.println("'" + entry.getKey() + "': " + entry.getValue());
        }
    }
}
```

5. Remove Duplicate Characters from a String

Write a program to remove duplicate characters from a string.

Example:

- Input: `"aabbcc"`
- Output: `"abc"`

```java
// Use this editor to write, compile and run your Java code online
import java.util.*;
class Main {
    public static void main(String[] args) {
        String str="lavanya";
        Set<Character> set = new LinkedHashSet<>();
        for(char c : str.toCharArray())
        {
         set.add(c);
        }
        StringBuilder sb = new StringBuilder();
        for(char c : set)
        {
         sb.append(c);
        }
         System.out.println("Duplicate Elements:"+sb);
    }
}
```

6. Find the nth non-repeating character in a string using Java

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        String str = "programming";
        int k=2;
        Map<Character, Integer> charCounts = new HashMap<>();
          for (int i=0;i<str.length();i++) {
            char c=str.charAt(i);
            charCounts.put(c, charCounts.getOrDefault(c, 0) + 1);
        }

        int nonRepeatingCount = 0;
        for(int i=0;i<str.length();i++){
            char c=str.charAt(i);
            if(charCounts.get(c) == 1){
                nonRepeatingCount++;
            if(nonRepeatingCount == k){
                System.out.println("Nth Non-repeating character is:"+c);}
        }
        }
    }
}
```

7. Check if Two Strings are Anagrams

Write a program to check if two strings are anagrams (contain the same characters in any order).

Example:

- Input: `"listen"`, `"silent"`
- Output: `true`

```java
import java.util.Arrays;

public class AnagramCheck {
    public static void main(String[] args) {
        String str1 = "listen";
        String str2 = "silent";
        char[] charArray1 = str1.toCharArray();
        char[] charArray2 = str2.toCharArray();
        Arrays.sort(charArray1);
        Arrays.sort(charArray2);
        if (Arrays.equals(charArray1, charArray2)) {
            System.out.println("The strings are anagrams.");
        } else {
            System.out.println("The strings are not anagrams.");
        }
    }
}
```

8. Remove Whitespace from a String

Write a program to remove all spaces or whitespace characters from a string.

Example:

- Input: `"Hello World"`
- Output: `"HelloWorld"`

```java
public class RemoveWhitespace {
    public static void main(String[] args) {
        String str = "Hello World";
        String result = str.replaceAll("\\s", "");
        System.out.println("String without spaces: " + result);
    }
}
```

9. Reverse the Words in a String

Write a program to reverse the words of a sentence.

Example:

- Input: `"Hello World"`
- Output: `"World Hello"`

```java
public class ReverseWords {
    public static void main(String[] args) {
        String str = "Hello World";
        String[] words = str.split(" ");
        StringBuilder reversed = new StringBuilder();
        for (int i = words.length - 1; i >= 0; i--) {
            reversed.append(words[i]).append(" ");
        }
        System.out.println("Reversed words: " + reversed.toString().trim());
    }
}
```

10. Check if a String is a Substring of Another String

Write a program to check if one string is a substring of another string.

Example:

- Input: `"hello world"`, Substring: `"world"`
- Output: `true`

```java
public class SubstringCheck {
    public static void main(String[] args) {
        String str1 = "hello world";
        String str2 = "world";
        if (str1.contains(str2)) {
            System.out.println(str2 + " is a substring of " + str1);
        } else {
            System.out.println(str2 + " is not a substring of " + str1);
        }
    }
}
```

11. Duplicate words in a string

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        String str = "programming is language is java";
       String[] words = str.toLowerCase().split("\\s+");
       Map<String, Integer> wordCountMap = new HashMap<>();
        for (String word : words) {
        wordCountMap.put(word, wordCountMap.getOrDefault(word, 0) + 1);
            }
        System.out.println("Duplicate words:");
        int duplicateWordCount = 0;
        for(Map.Entry<String, Integer> entry : wordCountMap.entrySet()) {
            if(entry.getValue() > 1) {
            duplicateWordCount++;
            System.out.println(entry.getKey() + ": " + entry.getValue());
                }
            }
    System.out.println("Number of duplicate words: "+duplicateWordCount);
    }
}
```

12. Print first letter of each word in a string in java

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        String sentence = "Learn with Krishna Sandhi";
        StringBuilder initials = new StringBuilder();

        String[] words = sentence.split(" ");
        for (String word : words) {
            if (!word.isEmpty()) {
                initials.append(word.charAt(0));
            }
        }

    System.out.println("The first letters are: " + initials.toString());
    }
}
```

13. Java program to count the number of words in a string

```
package test;

public class Test
{
    public static void main(String[] args)
    {
        String s = "Welcome to Java World ";
        int count = 1;
        for (int i = 0; i < s.length() - 1; i++)
        {
            if ((s.charAt(i) == ' ') && (s.charAt(i + 1) != ' '))
            {
                count++;
            }
        }
        System.out.println("Number of words in a string: " + count);
    }
}
```

```
Number of words in a string: 4
```

14. Java program to find all permutations of a given string

```
package test;

public class Test
{
    public static void main(String[] args)
    {
        String str = "abc";
        permute(str, "");
    }

    static void permute(String str, String prefix)
    {
        if (str.length() == 0)
        {
            System.out.println(prefix);
        }
        else
        {
            for(int i = 0; i < str.length(); i++)
            {
                String rem = str.substring(0,i) + str.substring(i + 1);
                permute(rem, prefix + str.charAt(i));
            }
        }
    }
}
```

```
abc
acb
bac
bca
cab
cba
```

15. Java program to print unqiue characters

```
public class Test
{
    public static void main(String[] args)
    {
        String str = "Java Automation ";
        boolean[] unique = new boolean[128];
        for (int i = 0; i < str.length(); i++)
        {
            char ch = str.charAt(i);
            if (!unique[ch])
            {
                unique[ch] = true;
                System.out.print(ch + " ");
            }
        }
    }
}
```

```
J a v   A u t o m i n
```

16. Java program to print even indexed characters

```
3  public class Test
4  {
5      public static void main(String[] args)
6      {
7          String str = "Automation ";
8          for (int i = 0; i < str.length(); i++)
9          {
10             if (i % 2 == 0)
11             {
12                 System.out.print(str.charAt(i));
13             }
14         }
15     }
16 }
17
```

Console × Problems Debug Shell

```
<terminated> Test [Java Application] C:\Users\Sunil\.p2\pool\plugins\org.eclipse.justj.openjdk
Atmto
```

17. Java program to print each letter twice from a given string

```java
package test;

public class Test
{
    public static void main(String[] args)
    {
        String str = "Automation";
        StringBuilder doubled = new StringBuilder();
        for (int i = 0; i < str.length(); i++) {
        char ch = str.charAt(i);
        doubled.append(ch).append(ch);
        }
        System.out.println("Doubled String:"+doubled);
    }
}
```

```
Console  Problems  Debug Shell
<terminated> Test [Java Application] C:\Users\Sunil\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32
Doubled String:AAuuttoommaattiioonn
```

18. Java program to swap two string without using 3rd variable

```java
package test;

public class Test {
    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);
    }
}
```

```
Console  Problems  Debug Shell
<terminated> Test [Java Application] C:\Users\Sunil\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin
Before swapping: str1 = hello, str2 = world
After swapping: str1 = world, str2 = hello
```

19. Java program to gives two Output: “abcde”, “ABCDE” for the Input String Str = “aBACbcEDed”

```java
package test;

public class Test {
    public static void main(String[] args) {
        String str = "aBACbcEDed";
        StringBuilder lowerCase = new StringBuilder();
        StringBuilder upperCase = new StringBuilder();
        for(char ch : str.toCharArray())
        {
        if(Character.isLowerCase(ch))
        {
        lowerCase.append(ch);
        }
        else
        {
        upperCase.append(ch);
        }
        }
        System.out.println("Output in lowercase: "+lowerCase);
        System.out.println("Output in uppercase "+upperCase);

    }
}
```

Console × Problems Debug Shell

<terminated> Test [Java Application] C:\Users\Sunil\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0

```
Output in lowercase: abced
Output in uppercase BACED
```

20. Java program to gives two Output: “Subburaj”, “123” for the Input String Str = “Subbu123raj”

```java
package test;

public class Test {
    public static void main(String[] args) {
        String str="Subbu123raj";
        StringBuilder alphaPart = new StringBuilder();
        StringBuilder numericPart = new StringBuilder();
        for(char ch : str.toCharArray())
        {
        if(Character.isLetter(ch))
        {
        alphaPart.append(ch);
        }
        else if (Character.isDigit(ch))
        {
        numericPart.append(ch);
        }
        }
        System.out.println("Output in Alpha: "+alphaPart.toString());
        System.out.println("Output in Numeric: " + numericPart.toString());

    }
}
```

Console × Problems Debug Shell

<terminated> Test [Java Application] C:\Users\Sunil\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v2025(

```
Output in Alpha: Subburaj
Output in Numeric: 123
```

21. Java program to gives Output: “32412120000” for the Input String Str = “32400121200”

```java
package test;

public class Test {
    public static void main(String[] args) {
        String input = "32400121200";
        StringBuilder digits = new StringBuilder();
        StringBuilder nonDigits = new StringBuilder();

        for (char c : input.toCharArray()) {
            if (Character.isDigit(c)) {
                digits.append(c);
            } else {
                nonDigits.append(c);
            }
        }
        String result = digits.toString() + nonDigits.toString();
        System.out.println("Output::" + result);
    }
}
```

Console × Problems Debug Shell

<terminated> Test [Java Application] C:\Users\Sunil\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_

```
Output::32400121200
```

22. Java program to gives Output: “00003241212” for the Input String Str = “32400121200”

```java
package test;

public class Test {
    public static void main(String[] args) {
        String input = "32400121200";
        String formattedOutput = String.format("%011d", Long.parseLong(input));
        System.out.println("Formatted output: " + formattedOutput);
    }
}
```

Console × Problems Debug Shell

<terminated> Test [Java Application] C:\Users\Sunil\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250

```
Formatted output: 32400121200
```

23. Java program to find the longest without repeating characters

```java
import java.util.HashSet;
public class Test {
    public int lengthOfLongestSubstring(String s) {
        int n = s.length();
        if (n == 0) {
            return 0;
        }
        Set<Character> charSet = new HashSet<>();
        int maxLength = 0;
        int left = 0;

        for (int right = 0; right < n; right++) {
            char currentChar = s.charAt(right);
            while (charSet.contains(currentChar)) {
                charSet.remove(s.charAt(left));
                left++;
            }
            charSet.add(currentChar);
            maxLength = Math.max(maxLength, right - left + 1);
        }
        return maxLength;
    }
    public static void main(String[] args) {
        Test solution = new Test();
        System.out.println(solution.lengthOfLongestSubstring("abcabcbb")); // Output: 3
        System.out.println(solution.lengthOfLongestSubstring("bbbbb")); // Output: 1
        System.out.println(solution.lengthOfLongestSubstring("pwwkew")); // Output: 3
        System.out.println(solution.lengthOfLongestSubstring("")); // Output: 0
        System.out.println(solution.lengthOfLongestSubstring("au")); // Output: 2
    }
}
```

Console Problems

<terminated> Test [Java Application

```
3
1
3
0
2
```