**Basic Level (1–10)**

1. Write a program to input a string and print its length.
2. Write a program to convert a string into uppercase and lowercase.
3. Write a program to count the number of vowels and consonants in a string.
4. Write a program to check whether a given string is a palindrome.
5. Write a program to reverse a string without using StringBuilder.reverse().
6. Write a program to compare two strings using == and .equals() and explain the output.
7. Write a program to count how many times a given character appears in a string.
8. Write a program to remove all white spaces from a string.
9. Write a program to find the first non-repeated character in a string.
10. Write a program to check whether two strings are anagrams of each other.

**Intermediate Level (11–20)**

1. Write a program to count the number of words in a string.
2. Write a program to find the largest and smallest word in a given string.
3. Write a program to find the frequency of each character in a string.
4. Write a program to replace all occurrences of a substring in a string with another substring.
5. Write a program to check whether one string is a rotation of another (e.g., "ABCD" and "CDAB").
6. Write a program to find all substrings of a given string.
7. Write a program to check whether a string contains only digits.
8. Write a program to remove duplicate characters from a string.
9. Write a program to count the number of uppercase, lowercase, digits, and special characters in a string.
10. Write a program to check if two strings are equal without using .equals().

**Advanced Level (21–30)**

1. Write a program to find the longest palindrome substring in a given string.
2. Write a program to sort the characters of a string in alphabetical order.
3. Write a program to check if two strings are isomorphic (characters can be replaced consistently).
4. Write a program to find the first repeated word in a string.
5. Write a program to print all permutations of a string.
6. Write a program to count the number of occurrences of each word in a string.
7. Write a program to check whether a string is a valid identifier in Java (no digits at start, no special symbols except \_, etc.).
8. Write a program to find the longest common prefix in a set of strings.
9. Write a program to implement a simple **String compression algorithm** (e.g., "aaabbc" → "a3b2c1").
10. Write a program to check if a string contains only unique characters (no duplicates).