**Unit-6 (String Handling)**

**String:**

A string is a sequence of characters surrounded by double quotations. In a java programming language, a string is the object of a built-in class **String**.

It is an **immutable** object, which means it can’t be updated once created.

The **String** class defined in the package **java.lang** package.

**Creating a String**  
There are two ways to create string in Java:  
**1. String literal**String s = “Swastik College”;

**2. Using new keyword**  
String s = new String (“Swastik College”);

### Concatenating String: There are 2 methods to concatenate two or more string. 1.)Using concat() method 2.)Using + operator

**Using concat() Method:  
concat()**method is used to add two or more string into a single string object. It is string class method and returns a string object.  
        String s = "Hello";

        String str = "Java";

        String str1 = s.concat(str);

        System.out.println(str1);

**Using + operator**

Java uses "+" operator to concatenate two string objects into single one. It can also concatenate numeric value with string object. See the below example.

        String s = "Hello";

        String str = "Java";

        String str1 = s+str;

        String str2 = "Java"+11;

        System.out.println(str1);

        System.out.println(str2);

### Conversion of String: convert a string to different data types or formats

### String to Integer: Use Integer.parseInt() method.

        String str = "123";

        int num = Integer.parseInt(str);

we can use **[Integer.valueOf()](https://www.javatpoint.com/java-integer-valueof-method)** method which returns instance of Integer class

        String s="200";

        //converting String into Integer using Integer.valueOf() method

        Integer i=Integer.valueOf(s);

### String to Double: Use Double.parseDouble() method.         String str = "3.14";         double num = Double.parseDouble(str);

### String to Boolean: Use Boolean.parseBoolean() method.

        String str = "true";

        boolean bool = Boolean.parseBoolean(str);

**String to Character Array: Use** toCharArray() **method.** String str = "hello";

        char[] charArray = str.toCharArray();

      for (char c : charArray) {

            System.out.println(c);

        }

String to Date: Use **SimpleDateFormat** or **DateTimeFormatter** (for Java 8 and later).

import java.text.SimpleDateFormat;

import java.util.\*;

public class App {

    public static void main(String[] args) throws Exception {

        String str = "2024-04-14";

        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

        Date date = dateFormat.parse(str);

        System.out.println(date);

    }

}

String to **LocalDateTime** (Java 8+):

import java.time.\*;

public class App {

    public static void main(String[] args) throws Exception {

        String str = "2024-04-14T12:00";

        LocalDateTime dateTime = LocalDateTime.parse(str);

        System.out.println(dateTime);

    }

}

## Exam Questions: Write any four String methods used in java with example. Important Java string methods:

|  |  |  |
| --- | --- | --- |
| **Method** | **Description** | **Return Type** |
| [charAt()](https://www.w3schools.com/java/ref_string_charat.asp) | Returns the character at the specified index (position)  String myStr = "Hello";  char result = myStr.charAt(0);  System.out.println(result);//h | char |
| [compareTo()](https://www.w3schools.com/java/ref_string_compareto.asp) | Compares two strings lexicographically  String myStr1 = "Hello";  String myStr2 = "Hello";  System.out.println(myStr1.compareTo(myStr2));  //0 | int |
| [concat()](https://www.w3schools.com/java/ref_string_concat.asp) | Appends a string to the end of another string  String firstName = "Sunil ";  String lastName = "Chaudhary";  System.out.println(firstName.concat(lastName)); | String |
| [contains()](https://www.w3schools.com/java/ref_string_contains.asp) | Checks whether a string contains a sequence of characters  String myStr = "Hello";  System.out.println(myStr.contains("Hel"));   // true  System.out.println(myStr.contains("e"));     // true  System.out.println(myStr.contains("Hi"));    // false | boolean |
| [endsWith()](https://www.w3schools.com/java/ref_string_endswith.asp) | Checks whether a string ends with the specified character(s)  String myStr = "Hello";  System.out.println(myStr.endsWith("Hel"));   // false  System.out.println(myStr.endsWith("llo"));   // true  System.out.println(myStr.endsWith("o"));     // true | boolean |
| [equals()](https://www.w3schools.com/java/ref_string_equals.asp) | Compares two strings. Returns true if the strings are equal, and false if not  String myStr1 = "Hello";  String myStr2 = "Hello";  String myStr3 = "Another String";  System.out.println(myStr1.equals(myStr2)); // Returns true because they are equal  System.out.println(myStr1.equals(myStr3)); // false | boolean |
| [format()](https://www.w3schools.com/java/ref_string_format.asp) | Returns a formatted string using the specified locale, format string, and arguments  String myStr = "Hello %s! One kilobyte is %,d bytes.";  String result = String.format(myStr, "World", 1024);  System.out.println(result); | String |
| [indexOf()](https://www.w3schools.com/java/ref_string_indexof.asp) | Returns the position of the first found occurrence of specified characters in a string  String myStr = "Hello planet earth, you are a great planet.";  System.out.println(myStr.indexOf("planet"));//6 | int |
| [join()](https://www.w3schools.com/java/ref_string_join.asp) | Joins one or more strings with a specified separator  String fruits = String.join(" ", "Orange", "Apple", "Mango");  System.out.println(fruits); | String |
| [lastIndexOf()](https://www.w3schools.com/java/ref_string_lastindexof.asp) | Returns the position of the last found occurrence of specified characters in a string          String myStr = "Hello planet earth, you are a great planet.";          System.out.println(myStr.lastIndexOf("planet"));//36 | int |
| [length()](https://www.w3schools.com/java/ref_string_length.asp) | Returns the length of a specified string  String txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";  System.out.println(txt.length());//26 | int |
| [replace()](https://www.w3schools.com/java/ref_string_replace.asp) | Searches a string for a specified value, and returns a new string where the specified values are replaced  String myStr = "Hello";  System.out.println(myStr.replace('l', 'p'));//heppo | String |
| [split()](https://www.w3schools.com/java/ref_string_split.asp) | Splits a string into an array of substrings  String str = "apple,banana,orange";  String[] fruits = str.split(",");  for (String fruit : fruits) {  System.out.println(fruit);  } | String[] |
| [startsWith()](https://www.w3schools.com/java/ref_string_startswith.asp) | Checks whether a string starts with specified characters          String myStr = "Hello";  System.out.println(myStr.startsWith("Hel"));//true  System.out.println(myStr.startsWith("llo"));//false  System.out.println(myStr.startsWith("o"));//false | boolean |
| [substring()](https://www.w3schools.com/java/ref_string_substring.asp) | Returns a new string which is the substring of a specified string  String myStr = "Hello, World!";  System.out.println(myStr.substring(7, 12));//World | String |
| [toCharArray()](https://www.w3schools.com/java/ref_string_tochararray.asp) | Converts this string to a new character array  String myStr = "Hello";  char[] myArray = myStr.toCharArray();  System.out.println(myArray[0]); | char[] |
| [toLowerCase()](https://www.w3schools.com/java/ref_string_tolowercase.asp) | Converts a string to lower case letters  String txt = "Hello World";  System.out.println(txt.toUpperCase());  System.out.println(txt.toLowerCase()); | String |
| [toString()](https://www.w3schools.com/java/ref_string_tostring.asp) | Returns the value of a String object | String |
| [toUpperCase()](https://www.w3schools.com/java/ref_string_touppercase.asp) | Converts a string to upper case letters  String txt = "Hello World";  System.out.println(txt.toUpperCase());//HELLO WORLD  System.out.println(txt.toLowerCase());//hello world | String |
| [trim()](https://www.w3schools.com/java/ref_string_trim.asp) | Removes whitespace from both ends of a string  String myStr = "       Hello World!       ";  System.out.println(myStr);  System.out.println(myStr.trim()); | String |
| [valueOf()](https://www.w3schools.com/java/ref_string_valueof.asp) | Returns the string representation of the specified value  char[] myArray = {'a', 'b', 'c'};  System.out.println(String.valueOf(myArray));//abc  System.out.println(String.valueOf('A'));//A  System.out.println(String.valueOf(true));//true  System.out.println(String.valueOf(4.5f));//4.5  System.out.println(String.valueOf(5.2));//5.2  System.out.println(String.valueOf(12));//12  System.out.println(String.valueOf(1400L));//1400 | String |

### Character Extraction: charAt(); substring(): toCharArray();6

### String Comparision: equals(): compareTo(); compareToIgnoreCase()

### Searching String: indexOf(): lastIndexOf(); contains(): startsWith():endsWith(): Modifying String: concat(): substring(): replace(): trim(): toLowerCase(): toUpperCase():

### Lab:Write a Java program that takes a string as input and counts the number of vowels (a, e, i, o, u) in it.

import java.util.\*;

public class App {

    public static void main(String[] args) throws Exception {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a string: ");

        String is = scanner.nextLine();

        scanner.close();

        is = is.toLowerCase();

       int vCount = 0;

       for (int i = 0; i < is.length(); i++) {

            char ch = is.charAt(i);

            if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {

                vCount++;

            }

        }

       System.out.println("Number of vowels: " + vCount);

    }

}

### Lab: Create a Java program that takes a sentence as input and capitalizes the first letter of each word.

public class App {

    public static void main(String[] args) throws Exception {

       String str="sunil chaudhary lahan";

       String[] strarr=str.split(" ");

       for (String s : strarr) {

        String stra="";

        for (int i = 0; i < s.length(); i++) {

            char c=s.charAt(i);

            if(i==0)

                stra=String.valueOf(c).toUpperCase();

            else

                stra+=String.valueOf(c);

        }

        System.out.print(stra+" ");

       }

     }

}

Lab: write a java Program to Accept a String and a Character and find out whether this characteris present in the string.If Present then display how many times this Character occurs

import java.util.\*;

public class App {

    public static void main(String[] args) throws Exception {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a string: ");

        String str = scanner.nextLine().toLowerCase();

        System.out.print("Enter a character: ");

        char ch = scanner.next().toLowerCase().charAt(0);

        // Count the occurrence of the character in the string

        int count = 0;

        for (int i = 0; i < str.length(); i++) {

            if (str.charAt(i) == ch) {

                count++;

            }

        }

        // Display the result

        if (count > 0) {

            System.out.println("present " + count + " times.");

        } else {

            System.out.println("is not present");

        }

        scanner.close();

     }

}

### String Buffer: the StringBuffer class is a mutable sequence of characters. It's similar to the String class, but it allows you to modify the contents of the string without creating a new object each time. This makes it more efficient when you need to perform a lot of string modifications.

### Difference between String and StringBuffer:

|  |  |  |
| --- | --- | --- |
| **S. No.** | **String** | **StringBuffer** |
| 1 | String is immutable. | It is mutable. |
| 2 | It is slow in terms of executing the concatenation task. | It is fast in terms of executing the concatenation task. |
| 3 | Here the length of the string class is static. | Here the length can be modified whenever required, as it is dynamic in behaviour. |
| 4 | It is less efficient. | It is more efficient in nature as compared to the string class. |
| 5 | String consumes more as compared to the stringbuffer. | StringBuffer uses less memory as compared to the string. |
| 5 | It utilises a string constant pool to store the values. | It prefers heap memory to store the objects. |
| 6 | It overrides both equal() and hashcode() techniques of object class. | It cannot override equal() and hashcode() methods. |

### Lab: Write a Java program to reverse a given string without using any built-in function.

public class App {

    public static void main(String[] args) throws Exception {

       String str="hello";

       int i=str.length();

        while (i>0)

        {

            System.out.println(str.charAt(i-1));

            i--;

        }

    }

}

Lab: Write a Java program to check whether a given string is a palindrome or not.

public class App {

    public static void main(String[] args) throws Exception {

       String str="liril";

       String mstr="";

       int i=str.length();

        while (i>0)

        {

            mstr +=str.charAt(i-1);

            i--;

        }

        if(str.equals(mstr))

            System.out.println("Palindrome");

        else

            System.out.println("Not Palindrome");

     }

}

Lab: Write a java program to find the repeated character in a string.

public class App {

    public static void main(String[] args) throws Exception {

        String input="Programming logic";

        char[] characters = input.toCharArray();

        int length = characters.length;

        System.out.println("Duplicate Characters: ");

        for (int i = 0; i < length; i++) {

            for (int j = i + 1; j < length; j++) {

                if (characters[i] == characters[j]) {

                    System.out.println(characters[i]);

                    break;

                }

            }

        }

     }

}

### 