String Class Methods

Method	Description	Return Type
charAt()	Returns the character at the specified index (position)	char
codePointAt()	Returns the Unicode of the character at the specified index	int
codePointBefore()	ırns the Unicode of the character before the specified index	int
codePointCount()	Returns the number of Unicode values found in a string.	int
compareTo()	Compares two strings lexicographically	int
compareToIgnoreCase()	Compares two strings lexicographically, ignoring case differences	int
concat()	Appends a string to the end of another string	String
contains()	Checks whether a string contains a sequence of characters	boolean
contentEquals()	Checks whether a string contains the exact same sequence of characters of the specified CharSequence or StringBuffer	boolean
copyValueOf()	Returns a String that represents the characters of the character array	String
endsWith()	Checks whether a string ends with the specified character(s)	boolean
equals()	Compares two strings. Returns true if the strings are equal, and false if not	boolean
equalsIgnoreCase()	Compares two strings, ignoring case considerations	boolean
format()	Returns a formatted string using the specified locale, format string, and arguments	String
getBytes()	Converts a string into an array of bytes	byte[]
getChars()	Copies characters from a string to an array of chars	void
hashCode()	Returns the hash code of a string	int
indexOf()	Returns the position of the first found occurrence of specified characters in a string	int
intern()	Returns the canonical representation for the string object	String
isEmpty()	Checks whether a string is empty or not	boolean

join()	Joins one or more strings with a specified separator	String
lastIndexOf()	Returns the position of the last found occurrence of specified characters in a string	int
length()	Returns the length of a specified string	int
matches()	Searches a string for a match against a regular expression, and returns the matches	boolean
offsetByCodePoints()	Returns the index within this String that is offset from the given index by codePointOffset code points	int
regionMatches()	Tests if two string regions are equal	boolean
replace()	Searches a string for a specified value, and returns a new string where the specified values are replaced	String
replaceAll()	Replaces each substring of this string that matches the given regular expression with the given replacement	String
replaceFirst()	Replaces the first occurrence of a substring that matches the given regular expression with the given replacement	String
split()	Splits a string into an array of substrings	String[]
startsWith()	Checks whether a string starts with specified characters	boolean
subSequence()	Returns a new character sequence that is a subsequence of this sequence	CharSequence
substring()	Returns a new string which is the substring of a specified string	String
toCharArray()	Converts this string to a new character array	char[]
toLowerCase()	Converts a string to lower case letters	String
toString()	Returns the value of a String object	String
toUpperCase()	Converts a string to upper case letters	String
trim()	Removes whitespace from both ends of a string	String
valueOf()	Returns the string representation of the specified value	String

Program

```
class Main {
   public static void main(String[] args) {
        {
      String s= "Welcome to Java Programming";
      // or String s= new String ("Welcome to Java Programming ");

      // Returns the number of characters in the String.
      System.out.println("String length = " + s.length());

      // Returns the character at ith index.
      System.out.println("Character at 3rd position = "
                  + s.charAt(3));

      // Return the substring from the ith  index character to end of string
      System.out.println("Substring " + s.substring(3));

      // Returns the substring from i to j-1 index.
      System.out.println("Substring  = " + s.substring(2,5));

      // Concatenates string2 to the end of string1.
      String s1 = "Good";
      String s2 = "Morning";
      System.out.println("Concatenated string  = " +
                   s1.concat(s2));
```

```
        // Returns the index within the string
String s4 = "I like Java ";

            System.out.println("Index of Java " +
                    s4.indexOf("Java"));

        // Returns the index within the string of the first occurrence of the specified string,
        // starting at the specified index.
        System.out.println("Index of a  = " +
                    s4.indexOf('a',1));

        // Checking equality of Strings
        Boolean out = "Java".equals("java");
        System.out.println("Checking Equality  " + out);

         Boolean outt = "Hello".equalsIgnoreCase("HELLO");
        outt = "Hello".equalsIgnoreCase("HELLO");
        System.out.println("Checking Equality " + outt);

        //If ASCII difference is zero then the two strings are similar

        int comp = (s1.compareTo(s2));
        if(comp==1)
        {
        System.out.println("The String "+" "+s1+" "+" is greater");
        }
```

```
        else
        {
            System.out.println("The String"+s2+"is greater");
        }

        // Converting cases
        String w1 = "  Hello!! Good Morning";
        System.out.println("Changing to lower Case " +
                        w1.toLowerCase());

        // Converting cases
        System.out.println("Changing to UPPER Case " +
                        w1.toUpperCase());

        // Trimming the word
        System.out.println("Trim the word " + w1.trim());

        // Replacing characters
        String str1 = "I love Java";
        System.out.println("Original String " + str1);
        String str2 = str1.replace('J' ,'j') ;
        System.out.println("Replaced J with j -> " + str2);
    }
}

    }
```