**Tutorial 6 String Class Methods**

**1)**

// Java code to illustrate different methods of String class.

import java.io.\*; import java.util.\*; class Test

{

public static void main (String[] args)

{

String s= "Bennett";

# // 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 = "Bennett"; String s2 = "University";

System.out.println("Concatenated string = " + s1.concat(s2));

# // Returns the index within the string of the first occurrence of the specified string.

String s4 = "Have Happy Hours";

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

# // Returns the index within the string of the

**// first occurrence of the specified string,**

**// starting at the specified index.**

System.out.println("Index of o = " + s4.indexOf('o',3));

# // Checking equality of Strings

Boolean out = "Happy".equals("happy"); System.out.println("Checking Equality " + out); out = "Happy".equals("Happy"); System.out.println("Checking Equality " + out);

# // Compares string to another string, ignoring case considerations.

out = "Happy".equalsIgnoreCase("hAppy");

System.out.println("Checking Equality" + out);

# // Compares two string lexicographically.

This returns difference s1-s2. If :

1. out < 0 // s1 comes before s2
2. out = 0 // s1 and s2 are equal.
3. out >0 // s1 comes after s2. int out1 = s1.compareTo(s2); System.out.println("If s1 = s2" + out);

# // Converts al the all characters in the String to lower case

String word1 = "HappyMe"; System.out.println("Changing to lower Case " +

word1.toLowerCase());

# // Converts al the all characters in the String to upper case

String word2 = "HappyME"; System.out.println("Changing to UPPER Case " +

word1.toUpperCase());

# // Replacing characters

String str1 = "havehappyhours"; System.out.println("Original String " + str1); String str2 = "havehappyhours".replace('h' ,'H') ; System.out.println("Replaced h with H -> " + str2);

}

}

# 2)

# One More Example

// Java code to illustrate different methods of String class.

import java.io.\*; import java.util.\*;

public class StringDemo

{

String s = "Bennett";

String s1 = "Live Long Life";

public int len(String k)

{

return k.length();

}

public String lowerCase()

{

return s.toLowerCase();

}

public String upperCase()

{

return s.toUpperCase();

}

public String addStrings(String add)

{

return s.concat(add);

}

public char wordAt(int i)

{

return s.charAt(i);

}

public String subSt(int j)

{

return s.substring(j);

}

public String subStt(int f, int e)

{

return s.substring(f,e);

}

public Boolean equalStrings(String a)

{

return s.equals(a);

}

public Boolean equalignoreStrings(String a)

{

return s.equalsIgnoreCase(a);

}

public int compareStrings(String add)

{

return s.compareTo(add);

}

public int lastIndexing(String a)

{

return a.lastindexOf('s');

}

/\*public String indexing(String add)

{

return s1.indexOf(add);

}

public String repl(char old, char new)

{

return s.replace(o , n);

}\*/

}

class Test

{

public static void main (String[] args)

{

StringDemo sd = new StringDemo();

String value;

int val;

val= sd.len("Monday"); System.out.println("Length is ::::" + val);

value=sd.lowerCase(); System.out.println("Lower Case ::::" + value);

value=sd.upperCase(); System.out.println("Upper Case ::::" + value);

String s2 = "University"; value=sd.addStrings(s2);

System.out.println("Concatenated Strings ::::" + value);

char a; a=sd.wordAt(6);

System.out.println("Character at given index ::::" + a);

value=sd.subStt(2,5); System.out.println("Substring ::::" + value);

Boolean y;

String s3="Bennett"; y= sd.equalStrings(s3);

System.out.println(" Strings are equal?? ::::" + y);

String s4="bennett"; y= sd.equalStrings(s4);

System.out.println(" Strings are equal?? ::::" + y);

y= sd.equalignoreStrings(s3); System.out.println(" Strings are equal?? ::::" + y);

y= sd.equalignoreStrings(s4); System.out.println(" Strings are equal?? ::::" + y);

val=sd.compareStrings(s2); System.out.println("Comparison of Strings ::::" + val);

String s6 = "Bennett"; val=sd.compareStrings(s6);

System.out.println("Comparison of Strings ::::" + val);

String s7 = "Beautiful"; val=sd.compareStrings(s7);

System.out.println("Comparison of Strings ::::" + val);

String s5= "Strings"; val=sd.lastIndexing(s5);

System.out.println(" last index of given Character ::::" + val); val= s5.lastindexOf('s')

// value=sd.indexing("Long");

// System.out.println("Concatenated Strings ::::" + value);

// value=sd.repl('B','S');

// System.out.println("After Replacement ::::" + value);

}

}

**3)**

Ques. Predict the output of the following code: 1)

**class** String\_demo

{

**public static void** main(String args[])

{

String s = “dcba”; String s1 = "abcd";

2)

**int** len1 = s1.length();

**int** len2 = s.length(); System.out.println(len1 + " " + len2);

}

}

**class** string\_demo

{

**public static void** main(String args[])

{

String obj = "I" + "like" + "Java"; System.out.println(obj);

}

}

3)

**class** string\_class

{

**public static void** main(String args[])

{

String obj = "I LIKE JAVA"; System.out.println(obj.charAt(3));

}

}

4)

**class** string\_class

{

**public static void** main(String args[])

{

String obj = "I LIKE JAVA"; System.out.println(obj.length());

}

}

5)

**class** string\_class

{

**public static void** main(String args[])

{

String obj = "hello"; String obj1 = "world"; String obj2 = obj; obj2 = " world";

System.out.println(obj + " " + obj2);

}

}

6)

**class** string\_class

{

**public static void** main(String args[])

{

String obj = "hello"; String obj1 = "world"; String obj2 = "hello";

System.out.println(obj.equals(obj1) + " " + obj.equals(obj2));

}

}

1. What is the output of the following code? String S;

System.Out.println(“S = “+S);

* 1. null
  2. error

C) = S

D) S =

1. Which of the following method call returns negative if S1<S2, positive if S1>S2 and zero if S1 is equal to S2.
   1. S2.compareTo(S1)
   2. S1.compareTo(S2)
   3. compare S1 To S2
   4. S1.compare(S2)
2. What will be the output? String s1 = "xy";

String s2 = s1;

s1 = s1 + s2 + "z";

1. xyz
2. xyxyz
3. xy xy z
4. xy z
5. z
6. What is the value of len after the following executes?

String s1 = "Hey, buddy!";

int len = s1.length();

1. 8
2. 10
3. 11