

Program to demonstrate Built-in functions of String Class **Code:**

```
1  import java.util.Scanner;
2  class StringFunctions
3  {
4      public static void main(String args[])
5      {
6          String str1 = "    Hello";
7          String str2 = "World";
8          String str3 = "From Java";
9          String str4 = "In Java";
10         String str5 = str3.concat(" ").concat(str4);
11         System.out.println("Returns 0 if str1 == \"Hello\": " +
12             ↳ str1.compareTo("Hello")); // compareTo()
13         System.out.println("str1 == \"Hello\": " + str1.equals("Hello"));
14             ↳ // equals()
15         System.out.println("str1 == \"Hello\" After trimming: " +
16             ↳ str1.trim().equals("Hello")); // trim()
17         System.out.println("str2.compareToIgnoreCase(\"world\") : " +
18             ↳ str2.compareToIgnoreCase("world")); // compareToIgnoreCase()
19         System.out.println("Compare str2.toLowerCase() and \"World\" ignoring
20             ↳ the case: " + str2.toLowerCase().equalsIgnoreCase("World")); //
21             ↳ toLowerCase() & equalsIgnoreCase()
22         System.out.println("str3.toUpperCase(): " + str3.toUpperCase()); //
23             ↳ toUpperCase()
24         System.out.println("Replace First occurrence of \"Java\" with \"Command
25             ↳ Prompt\": " + str5.replaceFirst("Java", "Command Prompt")); //
26             ↳ replaceFirst()
27         System.out.println("Replace All occurrences of \"Java\" with \"Command
28             ↳ Prompt\": " + str5.replaceAll("Java", "Command Prompt")); //
29             ↳ replaceAll()
30         System.out.println("Does str5.replaceAll(\"Java\", \"Command Prompt\")
31             ↳ contains \"Java\": " + str5.replaceAll("Java", "Command
32             ↳ Prompt").contains("Java")); // contains()
33         System.out.println("str5 ends with \"Java\": " + str5.endsWith("Java"));
34             ↳ // endsWith()
35         StringBuilder s = new StringBuilder(str3); // For using contentEquals
36             ↳ which takes a CharSequence parameter
37         System.out.println("Content of s equals content of str3: " +
38             ↳ str3.contentEquals(s)); // contentEquals()
39         System.out.println("\"\" is empty: " + "".isEmpty()); // isEmpty()
40         System.out.println("Replace first occurrence of I with 0 in str4: " +
41             ↳ str4.replace('I', '0')); // replace()
42         System.out.println("Length of str5: " + str5.length()); // length()
43         System.out.println("Character at index 3 in str5: " + str5.charAt(3));
44             ↳ // charAt()
```

```

27         System.out.println("Substring of str5 from the index of where \"Java\"
           ↳ is found: " + str5.substring(str5.indexOf("Java"))); // substring()
28     char arr[] = str1.toCharArray(); // toCharArray()
29     for(int i = 0; i < arr.length; i++)
30     {
31         if(arr[i] == ' ')
32         {
33             arr[i] = '_';
34         }
35     }
36     System.out.print("Replacing spaces with underscore in arr: ");
37     System.out.print(arr);
38 }
39 }

```

Output:

```

Returns 0 if str1 == "Hello": -40
str1 == "Hello": false
str1 == "Hello" After trimming: true
str2.compareToIgnoreCase("world"): 0
Compare str2.toLowerCase() and "World" ignoring the case: true
str3.toUpperCase: FROM JAVA
Replace First occurrence of "Java" with "Command Prompt": From Command Prompt In Java
Replace All occurrences of "Java" with "Command Prompt": From Command Prompt In Command Prompt
Does str5.replaceAll("Java", "Command Prompt") contains "Java": false
str5 ends with "Java": true
Content of s equals content of str3: true
"" is empty: true
Replace first occurrence of I with O in str4: On Java
Length of str5: 17
Character at index 3 in str5: m
Substring of str5 from the index of where "Java" is found: Java In Java
Replacing spaces with underscore in arr: ____Hello

```

Matrix Class:

```

1  // matrix/Matrix.java
2  package matrix;
3  import java.util.Scanner;
4  public class Matrix
5  {
6      int arr[][];
7      int rows, columns;
8      public Matrix(int rows, int columns)
9      {
10         arr = new int[rows][columns];
11         this.rows = rows;
12         this.columns = columns;

```

```

13     }
14     public Matrix()
15     {
16         arr = new int[2][2];
17         rows = 2;
18         columns = 2;
19     }
20     public int elementAt(int row, int column)
21     {
22         return arr[row][column];
23     }
24     public void setElement(int row, int column, int data)
25     {
26         arr[row][column] = data;
27     }
28     public void setMatrix()
29     {
30         Scanner sc = new Scanner(System.in);
31         for(int i = 0; i < rows; i++)
32         {
33             for(int j = 0; j < columns; j++)
34             {
35                 System.out.print("mat[" + i + "]" + "[" + j + "]: ");
36                 this.setElement(i, j, sc.nextInt());
37             }
38         }
39     }
40     public String toString()
41     {
42         StringBuilder str = new StringBuilder();
43         for(int i=0; i < rows; i++)
44         {
45             for(int j = 0; j < columns; j++)
46             {
47                 str.append(this.elementAt(i, j));
48                 str.append(' ');
49             }
50             str.append('\n');
51         }
52         return str.toString();
53     }
54
55     public Matrix transpose()
56     {
57         Matrix matTranspose = new Matrix(rows, columns);
58         for(int i = 0; i < rows; i++)
59         {

```

```

60         for(int j = 0; j < columns; j++)
61         {
62             matTranspose.setElement(i, j, this.elementAt(j, i));
63         }
64     }
65     return matTranspose;
66 }
67
68 public boolean equals(Matrix mat)
69 {
70     if( rows != mat.columns || columns != mat.columns)
71     {
72         System.out.print("Cannot Compare these matrices");
73     }
74     for(int i = 0; i < rows; i++)
75     {
76         for(int j = 0; j < columns; j++)
77         {
78             if(this.elementAt(i, j) != mat.elementAt(i, j))
79             {
80                 return false;
81             }
82         }
83     }
84     return true;
85 }
86
87 public int getColumns()
88 {
89     return columns;
90 }
91
92 public int getRows()
93 {
94     return rows;
95 }
96 }

```

To check if the entered matrix is symmetric or not **Code:**

```

1  // Symmetric.java
2  import java.util.Scanner;
3  import matrix.Matrix;
4  class Symmetric
5  {
6      static boolean isSymmetric(Matrix mat)
7      {
8          return mat.equals(mat.transpose());

```

```

9      }
10     public static void main(String args[])
11     {
12         Scanner sc = new Scanner(System.in);
13         System.out.print("Enter the order of the Matrix: ");
14         int order = sc.nextInt();
15         Matrix mat2 = new Matrix(order, order);
16         for(int i = 0; i < order; i++)
17         {
18             for(int j = 0; j < order; j++)
19             {
20                 System.out.print("mat[" + i + "]" + "[" + j + "]: ");
21                 mat2.setElement(i, j, sc.nextInt());
22             }
23         }
24         System.out.println(mat2);
25         System.out.println("The Matrix is " + ((isSymmetric(mat2) ? "Symmetric"
26             ↪ : "Not Symmetric")));
27     }
28     // SampleClass.java
29     import package1.*;
30     import package2.ClassA;
31     import package2.packageA.*;
32     class SampleClass
33     {
34         public static void main(String args[])
35         {
36             package1.Class1 c1 = new package1.Class1();
37             Class2 c2 = new Class2();
38             Class3 c3 = new Class3();
39             ClassA cA = new ClassA();
40             package2.packageA.Class1 c31 = new package2.packageA.Class1();
41             System.out.println("hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi
42             ↪ hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi
43             ↪ hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi
44             ↪ hi hi");
45             {
46                 {
47                     System.out.println("hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi
48                     ↪ hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi
49                     ↪ hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi");
50                 }
51             }
52         }
53     }

```

Output:

Enter the order of the Matrix: 3

mat[0][0]: 1

mat[0][1]: 0

mat[0][2]: 0

mat[1][0]: 0

mat[1][1]: 1

mat[1][2]: 0

mat[2][0]: 0

mat[2][1]: 0

mat[2][2]: 1

1 0 0

0 1 0

0 0 1

The Matrix is Symmetric

To Perform Matrix Multiplication **Code:**

```
1 // Multiplication.java
2 import java.util.Scanner;
3 import matrix.Matrix;
4 class Multiplication
5 {
6     static Matrix multiply(Matrix mat1, Matrix mat2)
7     {
8         if(mat1.getColumns() != mat2.getRows())
9         {
10             System.out.println("Cannot Multiply these matrices");
11         }
12         Matrix matMult = new Matrix(mat1.getRows(), mat2.getColumns());
13         for(int i=0; i < mat1.getRows(); i++)
14         {
15             for(int j=0; j < mat1.getColumns(); j++)
16             {
17                 for(int k = 0; k < mat2.getRows(); k++)
18                 {
19                     matMult.setElement(i, j, matMult.elementAt(i, j) +
↪ mat1.elementAt(i, k) * mat2.elementAt(k, j));
20                 }
21             }
22         }
23         return matMult;
24     }
25     public static void main(String args[])
26     {
27         Scanner sc = new Scanner(System.in);
```

```

28     System.out.print("Enter the no of rows of Matrix1: ");
29     int rows = sc.nextInt();
30     System.out.print("Enter the no of columns of Matrix1: ");
31     int columns = sc.nextInt();
32     Matrix mat1 = new Matrix(rows, columns);
33     mat1.setMatrix();
34     System.out.print("Enter the no of rows of Matrix2: ");
35     rows = sc.nextInt();
36     System.out.print("Enter the no of columns of Matrix2: ");
37     columns = sc.nextInt();
38     Matrix mat2 = new Matrix(rows, columns);
39     mat2.setMatrix();
40     System.out.println("mat1 x mat2 = ");
41     System.out.print(multiply(mat1, mat2));
42 }
43 }

```

Output:

```

Enter the no of rows of Matrix1: 2
Enter the no of columns of Matrix1: 2
mat[0][0]: 1
mat[0][1]: 1
mat[1][0]: 1
mat[1][1]: 1
Enter the no of rows of Matrix2: 2
Enter the no of columns of Matrix2: 2
mat[0][0]: 1
mat[0][1]: 2
mat[1][0]: 1
mat[1][1]: 2
mat1 x mat2 =
2 4
2 4

```

Reverse the string and decide whether it is palindrome or not and Capitalize the String **Code:**

```

1  import java.util.Scanner;
2  class Pallindrome
3  {
4      public static void main(String args[])
5      {
6          Scanner sc = new Scanner(System.in);
7          System.out.print("Enter a String: ");
8          String in = sc.next();
9          char str[] = in.toCharArray();
10         char rev[] = new char[str.length];
11         for(int i = 0; i < str.length; i++)
12         {

```

```
13         rev[i] = str[str.length - 1 - i];
14     }
15     System.out.println("The String is " + (in.equals(new String(rev)) ?
    ↪    "Pallindrome" : "Not Pallindrome"));
16     System.out.println("Capitalized String: " + in.toUpperCase());
17 }
18 }
```

Output:

```
Enter a String: naman
The String is Pallindrome
Capitalized String: NAMAN
```