

String Handling

String Class and its Methods

1. What is the **String**?

The **String** is a **library class** (predefined class) and it is used as a **data type**.

E.g.: **String** a = "Radha"; Here **String** is the class.

2. What kind of **data type** is **String**?

String is a **reference data type** because it is a built-in **class**. Class is one of the reference data types.

3. What is meant by **String objects**?

String variables and **String values** are known as **String objects**. E.g.: String a = "Radha"; Here **a** is the variable object, "**Radha**" is the value object.

4. What is the **use of String class methods**?

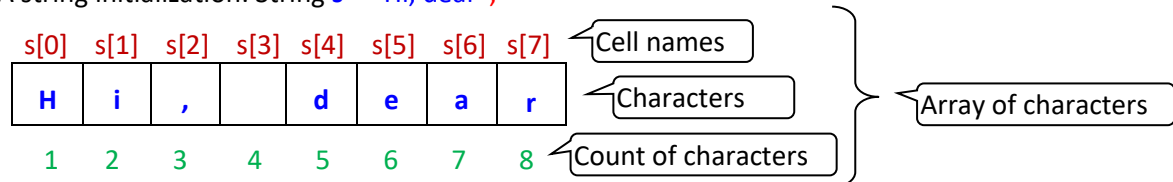
Methods of String class are used to solve problems on string values i.e., **words** and **sentences**.

5. In which **package** the String class resides? What is the **peculiarity** of this **package**?

The String class resides in **java.lang** package. it is a **default package**. It means this package **need not to be imported**.

Diagrammatic Representation of a String Object

A string initialization: String **s** = "Hi, dear";



String is an **array of characters**. Each character remains in each cell. The object **s** has **8** cells. (Number of cells is known as **length**. So length of **s** is **8**. Letter **l** is used as variable for the **length**.)

The **cells are named** as s[0] s[1] s[2] s[3] s[4] s[5] s[6] s[7].

0 1 2 3 4 5 6 7 are **indices (subscripts)**. The **index** starts from **0**.

The **last index** of a strings which has **l** characters will be **l-1**.

Each cell contains a character. **Special characters** and **spaces** are also considered as characters.

To store **special characters** and **spaces** it also requires individual cells.

Some Most Used String Methods (In Alphabetical Order)

Method Prototype

1) **char** charAt(**int** **index**)

Description

Returns the character at a specified index of a calling String object.

0123

char c = "Dear".charAt(**2**);

Returns **a** (i.e., the value of c is **a**)

E.g.: System.out.println("Dear".charAt(**2**));

The output is **a**

Note: The object and argument can be a variable also. E.g.:

String **s** = "Dear";

int **n** = 2;

char c = **s**.charAt(**n**);

Returns **a**

2) **int** compareTo(**String** **s**)

Compares two strings lexicographically (alphabetical order).

Returns an integer **>0** if **first one is greater** than second. Returns an integer **<0** if **first one is smaller** than second. Returns **0** if **both are same**. It is **case sensitive**. The return value is the difference between ASCII values of first or following characters of the strings.

int n = "**Binu**".compareTo("**Anoop**");

Returns **1**

int n = "**Anoop**".compareTo("**Ciya**");

Returns **-2**

int n = "**Anoop**".compareTo("**Anoop**");

Returns **0**

	<pre>int n = "anoop".compareTo("Anoop"); Returns 32 int n = "Anoop".compareTo("anoop"); Returns -32 int n = "Achu".compareTo("Abdhul"); Returns 1 int n = "Abdhul".compareTo("Abel"); Returns -1 int n = "Abhinav".compareTo("Abhi"); Returns 3 (7-4) difference of lengths int n = "Abhi".compareTo("Abhinav"); Returns -3 (4-7) difference of lengths.</pre>
3) <i>int</i> compareToIgnoreCase(<i>String s</i>)	Same as above, but it is case insensitive. <pre>int n = "anoop".compareToIgnoreCase("Anoop"); Returns 0 int n = "Anoop".compareToIgnoreCase("Binu"); Returns -1 int n = "Binu".compareToIgnoreCase("Anoop"); Returns 1</pre>
4) <i>String</i> concat(<i>String s</i>)	Concatenates the argument string to the end of a calling String object. (Strings only). <pre>String s = "Sree".concat("krishna"); Returns Sreekrishna</pre>
+ (concatenate operator)	An alternative to concat() method. But the + concatenates not only two strings but it concatenates all primitive type values to a string. <pre>String s = "Sum =" + ' ' + (5+3)+6.5; Returns Sum = 86.5</pre>
5) <i>boolean</i> endsWith(<i>String s</i>)	Checks whether a String ends with a specified substring. Returns boolean value true if it is true else false. <pre>boolean x = "Teacher".endsWith("her"); Returns true boolean x = "Teacher".endsWith("Tea"); Returns false boolean x = "Teachers".endsWith("her"); Returns false boolean x = "Teacher".endsWith("Teacher"); Returns true</pre>
6) <i>boolean</i> equals(<i>String s</i>)	Checks whether two strings are equal or not; considering case (upper and lower). Returns boolean value true if both are equal else false. <pre>boolean x = "Anu".equals("Binu"); Returns false boolean x = "Anu".equals("Anu"); Returns true boolean x = "Anu".equals("anu"); Returns false</pre>
7) <i>boolean</i> equalsIgnoreCase(<i>String s</i>)	Checks whether two strings are equal or not; ignoring case consideration. Returns boolean value true if both are equal else false. <pre>boolean x = "Anu".equalsIgnoreCase("anu"); Returns true boolean x = "Anu".equalsIgnoreCase("Anu"); Returns true boolean x = "Anu".equalsIgnoreCase("Binu"); Returns false</pre>
8) <i>int</i> indexOf(<i>char ch</i>)	Returns index of first occurrence of a specified character in a string. <pre>0123456 int n = "Nandana".indexOf('a'); Returns 1 0123456 int n = "Anandan".indexOf('a'); Returns 2 int n = "Nandana".indexOf('x'); Returns -1 (The -1 indicates there is no such a character.)</pre>

- 9) *int* indexOf(*char ch*, *int n*) Returns index of **first occurrence** of a specified character in a string **from a specified index**.
 0123456
int n = "N**and**ana".indexOf('a',2); Returns **4**
 0123456
int n = "N**and**ana".indexOf('a',1); Returns **1**
 0123456
int n = "N**and**ana".indexOf('a',7); Returns **-1**
- 10) *int* lastIndexOf(*char ch*) Returns index of **last occurrence** of a specified character in a string.
 01234567
int n = "Nandan**a**n".lastIndexOf('a'); Returns **6**
- 11) *int* lastIndexOf(*char ch*, *int n*) Returns the index of **last** occurrence of a specified character in a string before or the specified index.
 01234567
int n = "Nandan**a**nan".lastIndexOf('a',5); Returns **4**
 01234567
int n = "Nandan**a**n".lastIndexOf('a',6); Returns **6**
- 12) *int* length() Returns the length of a calling string.
 (Length means number of characters.)
 1234 (The count starts from 1)
int l = "Dear".length();
 Returns **4**
- 13) *String* replace(*char oldc*, *char newc*) All occurrence of a specified character is replaced with a new character and returns the modified string.
String a = "Ran**j**ini".replace('i', 'a');
 Returns Ran**j**ana
- 14) *String* replace(*String olds*, *String news*) All occurrence of a specified substring is replaced with a new substring and returns the modified string.
String a = "It **was** a working day but there **was** no class";
a = *a*.replace("was", "is");
 Returns It **is** a working day but there **is** no class
- 15) *boolean* startsWith(*String s*) Checks whether a String starts with a specified substring. Returns boolean value **true** if it is true else **false**.
boolean x = "Teacher".startsWith("Tea");
 Returns **true**
boolean x = "Teacher".startsWith("tea");
 Returns **false** (case sensitive)
boolean x = "Teacher".startsWith("her");
 Returns **false**
- 16) *String* substring(*int index*) Returns a substring from a specified index (**inclusive**) to the end of the calling String object.
 0123456
String s = "Fr **i**ends".substring(3);
 Returns **ends**
- 17) *String* substring(*int index1*, *int index2*) Returns a substring from index1 (**inclusive**) to index2 (**exclusive**) of calling String object.
 0123456
String s = "T**e**acher".substring(1,5);
 Returns **each**
- 18) *String* toLowerCase() Converts all of the characters of calling String object to lower case.
String a = "HELLO".toLowerCase();
 Returns **hello**

19) <i>String</i> toUpperCase()	Converts all of the characters of calling String object to upper case. String a = "hello".toUpperCase(); Returns HELLO
20) <i>String</i> trim()	Removes white spaces from both ends of the calling String object String a = " Hello Arun "; System.out.println(a.length()); a=a.trim(); System.out.println(a.length()); Output: 12 (Note there are two spaces at both ends) 10 (Two spaces are removed from both ends)
21) <i>String</i> valueOf(<i>any type</i>)	Returns string representation of the passed argument. e.g., integer 15 as string 15. int a=15,b=3; String c = "" .valueOf(a);//The null string object "" is used to call valueOf() . System.out.println(a+b); System.out.println(c+b); Output: 18 (a and b are integers. So adds) 153 (c is a string. So concatenates)

Conceptual Questions

Write output of the followings:

1. `char c="computer".charAt(3);`
2. `int x="Anna".compareTo("anna");`
3. `int x="Anna".compareToIgnoreCase("anna");`
4. `String a= "Aby",b= "Jose";`
`a=a.concat(b);`
`System.out.println(a);`
5. `String a= "Aby",b= "Jose";`
`System.out.println(a.concat(b));`
`System.out.println(a);`
6. `System.out.println("Dia".concat(" ").concat("Nair"));`
7. `boolean s = "Dear friend".endsWith("end");`
8. `boolean x="shilpa".equals("Shilpa");`
9. `boolean x = "Shilpa".equalsIgnoreCase("shilpa");`
10. `int n="Anaswara".indexOf('a');`
11. `int n="anaswara".indexOf('a',2);`
12. `int n="anaswara".indexOf('a',3);`
13. `int n="anaswara".lastIndexOf('a',1);`
14. `int n="anandan".lastIndexOf('a');`
15. `int n="anandan".lastIndexOf('a',4);`
16. `int n="anandan".lastIndexOf('a',5);`
17. `int c="computer".indexOf("computer".charAt(0));`
18. `int c="computer".indexOf("program".charAt(2));`
19. `int c="computer".charAt("program".indexOf('o'));`
20. `char c="anandan".charAt("program".indexOf('r',2));`
21. `char c="anjana".charAt("ranjana".indexOf('a',1));`
22. `char c = "nandana".charAt("anandan".lastIndexOf('a',4));`
23. `char c = "ranjini".charAt("ani".length());`
24. Statement to print middle character of a word in `s` where the length is an odd number.
25. `String c = "ranjini".replace('i', 'a');`
26. `String c = "renjini".replace('a', 'i');`
27. `String s="Jithu and Jose and Toby".replace("and", "or");`
28. `boolean n = "Gouri".startsWith("Go");`
29. `boolean n = "Gokul".endsWith("Go");`
30. `String s = "friend".substring(3);`
31. `String c = "Haripriya".substring(0,4);`
32. `String c= "ann".substring(0,1).toUpperCase();`
33. `String s= "Reshma".substring(0,1).toLowerCase();`
34. `String s = " Dear Dia ";`
`System.out.println(s.length());`
`s=s.trim();`
`System.out.println(s.length());`
35. `int a=5,b=20;`
`String c="".valueOf(a);`
`System.out.println(a+b);`
`System.out.println(b+c);`

Answer

Write output of the followings:

1. char c="computer".charAt(3); p
2. int x="Anna".compareTo("anna"); -32
3. int x="Anna".compareToIgnoreCase("anna"); 0
4. String a= "Aby",b= "Jose";
a=a.concat(b);
System.out.println(a); AbyJose
5. String a= "Aby",b= "Jose";
System.out.println(a.concat(b)); AbyJose
System.out.println(a); Aby
6. System.out.println("Dia".concat(" ").concat("Nair")); Dia Nair
7. boolean s = "Dear friend".endsWith("end"); true
8. boolean x="shilpa".equals("Shilpa"); false
9. boolean x = "Shilpa".equalsIgnoreCase("shilpa"); true
10. int n="Anaswara".indexOf('a'); 2
11. int n="anaswara".indexOf('a',2); 2
12. int n="anaswara".indexOf('a',3); 5
13. int n="anaswara".lastIndexOf('a',1); 0
14. int n="anandan".lastIndexOf('a'); 5
15. int n="anandan".lastIndexOf('a',4); 2
16. int n="anandan".lastIndexOf('a',5); 5
17. int c="computer".indexOf("computer".charAt(0)); 0
18. int c="computer".indexOf("program".charAt(2)); 1
19. char c="computer".charAt("program".indexOf('o')); m
20. char c="anandan".charAt("program".indexOf('r',2)); d
21. char c="anjana".charAt("ranjana".indexOf('a',1)); n
22. char c = "nandana".charAt("anandan".lastIndexOf('a',4)); n
23. char c = "ranjini".charAt("ani".length()); j
24. Statement to print middle character of a word in s. System.out.println(s.charAt(s.length()/2);
25. String c = "ranjini".replace('i','a'); ranjana
26. String c = "renjini".replace('a','i'); renjini
27. String s="Jithu and Jose and Toby".replace("and","or"); Jithu or Jose or Toby
28. boolean n = "Gouri".startsWith("Go"); true
29. boolean n = "Gokul".endsWith("Go"); false
30. String s = "friend".substring(3); end
31. String c = "Haripriya".substring(0,4); Hari
32. String c= "ann".substring(0,1).toUpperCase(); A
33. String s= "Reshma".substring(0,1).toLowerCase(); r
34. String s = " Dear Dia ";
System.out.println(s.length()); 10
s=s.trim();
System.out.println(s.length()); 8
35. int a=5,b=20;
String c="".valueOf(a);
System.out.println(a+b); 25
System.out.println(b+c); 205