

Topic: String handling in Java
Subject-Computer Application, Class-X

Date-13.05.2024
Teacher-Prabdeep Kaur

11. Substring()

~~Syntax~~
String substring = string.substring(start index);

* String s = "Tender Heart" substring(8)
s = Heart

* String substring = string.substring(start index, last index)
String s = "Tender Heart" substring(2, 8)
s = nder H
excluding last

12. concat()

~~Syntax~~ String variable = string.concat(string2)

String s1 = "Tender".concat("Heart");
s1 = TenderHeart
only join two strings

String s1 = "Leaf" + "Flowers" + "Plant"
to join more than two strings
s1 = LeafFlowersPlant

13. replace ()

return type = String

Syntax

1. String variable = String.replace(old char, new char);

String S = "String".replace('N', 'M')

String

replace all character
of string wherever n is with m

2. String variable = String.replace(old sequence of
Char, new sequence of char);

String S = "This is a book".replace
("is", "with")

This with a book.

replace sequence of character
also.

String handling in java

14. endsWith()

return type is boolean

```
boolean variable = String.endsWith(string2);
```

```
boolean ans = "Tender heart high School".endsWith  
              ("School");  
ans = True
```

```
boolean ans = "Tender Heart high School"  
              .endsWith("Chandigarh");  
ans = False
```

15. startsWith()

return type is boolean

```
boolean variable = String.startsWith(string2);
```

```
boolean ans = "Tender heart".startsWith  
              (Tender)
```

```
boolean ans = "Tender heart".startsWith  
              (School)
```

ans = False

Programs

WAP to accept a name and display the initials along with the Surname

Sample Input:

```
import java.util.*;
```

```
class init
```

```
{
```

```
    public static void main
```

```
    {
```

```
        Scanner sc = new Scanner (System.in)
```

```
        System.out.println("Enter name");
```

```
        String s = sc.nextLine();
```

```
        s = ' ' + s; // add the space
```

```
        l = s.length();
```

```
        l1 = s.lastIndexOf(' ');
```

```
        String s1 = s.substring(l1);
```

```
        for(int i = 0 ; i < l1 ; i++)
```

```
        {
```

```
            char ch = s.charAt(i);
```

```
            if(ch == ' ')
```

```
                String s2 = s2 + s.charAt(i+1) + " . " ;
```

```
        }
```

```
        String s3 = s2 + s1 ;
```

```
        // String s3 = s2.concat(s1);
```

```
        System.out.println(s3);
```

```
    }
```

```
}
```

Class: X

Subject: Computer Application

Topic: String Handling

Teacher: Prabdeep

Good Morning Students

Students, this lesson is of Class X for the subject of Computer Application. Today we start with our new topic "String Handling". Let start with the topic.

String Class

A string containing digits within double quotes is treated as a symbol and hence cannot be used for an arithmetical calculations.

JCL (Java Class Library) contains a class called String which is available under java.lang package. It allows the user to create an object or declare a variable to store a set of characters. Example "COMPUTER", "6479" etc.

String variable

String variable is a named memory location that contains a string constant or string literal. The value of variable may change in the program. A variable declared with data type string gains the capability of storing a set of characters.

Assigning a string : A string can be assigned in two different ways :

1. Direct Assignment: In this method, a string constant is directly assigned to a string variable. Syntax : `<string data type> variable = <string literal>`

Example: `String st = "understanding";`
`String st = "computer is fun";`

Class: X

Subject: Computer Application

Topic: String handling

Teacher: Prabhdeep

2. Assignment by using new keyword: It allows the user to assign a String object with a String literal/variable by using new keyword with the help of a Constructor. Syntax:

<class name> <object name> = new <constructor>
<string literal/variable>;

Example: String str = new String("understanding");
It means the string object str is assigned with a String literal "understanding".

By default, String data type creates a character array to store a set of characters in different cells. So, assigning a string means assigning a character array to a String object.

STRING FUNCTIONS: Like Math and character classes, String class also possesses some useful functions to operate or manipulate String data. These functions are listed below:

1. int length(): This function is used to return the length of a string that means number of characters present in the string. Note that the length of the string will be an integer number. So it return the integer value.

Syntax: <int variable> = <String.length>;

Example: String str = "COMPUTER";
int k = str.length();

Here, k = 8, as the number of characters in the given string str is 8.

Class: X

Subject: Computer Application

Topic: String handling

Teacher: Prabhdeep

2. Char charAt(int index): This function returns a character from the given index (position number) of the string.

Syntax: `<char variable> = <String charAt(int index)>;`

Example: `String s = "Computer";`

`char c = s.charAt(3);`

Here, `c = 'P'` as the character at 3rd index is p.

3. int indexOf(char ch): `indexOf()` function is opposite to the `charAt()` function. It returns the index that is the position number of first occurrence of a character in the string.

Syntax: `<int variable> = <String.indexOf(character)>;`

`String st = "underStanding";`

`int p = st.indexOf('N');`

Here, the return value for the variable `p` is 1 because the first occurrence of 'N' in the string is 1.

4. int lastIndexOf(char ch): This function is used to find the index of the last occurrence of a character in the string. Syntax: `<int variable> = <String lastIndexOf(character)>;`

Example: `String s = "MALAYALAM";`

`int n = s.lastIndexOf('A');`

It returns 7 to the integer variable `n`.

Hence, `n = 7`.

5. String substring(int start_index): This function is used to extract a set of characters simultaneously from a given index up to end of the string.

Syntax: `<String variable> = <String.substring(int start_index)>;`

Class: X

Subject: Computer Application

Topic: String handling

Teacher: Rabbhdeepkaur

Example: String s = "COMPUTER";

String p = S.substring(3);

The function will return all the characters of the String starting from the 3rd index till end. Therefore, P = PUTER. As the function will return a set of characters, hence, we have to use a string variable for storing the result.

6. String Substring(int start_index, int last_index): It is the another modification of substring function. It will return a set of characters simultaneously from start index up to the last index of the string, excluding the character present at the last index.

Syntax: <String variable> = <String Substring (int start_index, int last_index)>;

Example: String s = "COMPUTER";

String p = S.substring(3, 6)

It will return a set of characters of the string from the 3rd index up to the 6th index by excluding the character at the 6th index. Hence, p will result in PUT.

7. String toLowerCase(): This function converts a given string into lower case characters. If any character is already in lower case or is a special character then it will remain same.

Syntax: <String variable1> = <String toLowerCase()>;

8. String toUpperCase(): This function is opposite to the toLowerCase() function. It is used to convert a string into upper case characters.

Syntax: <String variable1> = <String toUpperCase()>;

Class: X

Subject: Computer Application

Topic: String handling

Teacher: Prabhdeep

8. String replace(char old_char, char new_char):

This function is used to replace a character by another character or to replace a string by another string at all its occurrences in the given string. Syntax: <String variable> = <String.replace(character to replace, character to appear)>;

9. String concat(String s): The concat() function is used to concatenate (Join) two strings together. Syntax: <String variable> = <String 1.concat(String 2)>;

10. boolean equals(String s): This function is used to compare two strings together to check whether they are identical or not. It returns a boolean type value true if both are same, otherwise false.

Syntax: <boolean variable> = <String 1.equals(String 2)>;

11. String equalsIgnoreCase(String s): This function is also used to compare two strings to check whether both are identical or not after ignoring the case that is corresponding upper and lower case characters are treated to be the same. It also returns a boolean type value as true or false. Syntax: <boolean variable> = <String 1.equalsIgnoreCase(String 2)>;

12. int compareTo(String s): It is also a function in java language package to compare two strings. It not only checks the equality of two strings but also checks whether a string is bigger or smaller than the other. It will return 0 if both the strings are same, a negative value if first string is smaller and

Class: X

Subject: Computer Application

Topic: String handling

Teacher: Prabdeep

a positive value if the first string is bigger.

Syntax: $\langle \text{int variable} \rangle = \langle \text{String 1 compareTo}(\text{String 2}) \rangle;$

The value of variable = 0 means both the strings are same.

The value of variable > 0 means String 1 is bigger than String 2.

The value of variable < 0 means String 1 is less than String 2.

13. int compareToIgnoreCase(String s): It is another modification of compareTo() function. It is used to perform the same task which above mentioned function does. Only difference is that the compareToIgnoreCase() function compares two strings after ignoring the cases that is corresponding upper case and lower case characters of the strings are treated to be same. Syntax: $\langle \text{int variable} \rangle = \langle \text{String 1.compareToIgnoreCase}(\text{String 2}) \rangle;$

14. String trim(): This function is used to remove leading and trailing blanks from a string. The other blanks, which are available in between the words will remain unchanged.

Syntax: $\langle \text{String variable} \rangle = \langle \text{String.trim}() \rangle;$

With this student I am ending the assignment and we have finish all term II syllabus also. Next assignments are only about your programs and term II exams.