Lecture Notes
K.Yellaswamy
Assistant Professor
K L University

Building Strings and Exploring String Class:

The String class

String: A String is a sequence of characters.

ex:- name,address,creditcard no,etc..,
In java,Any string is object of String class.
->String is not a character array.

The String class has 11 constructors and more than 40 methods.

create String object:-

1.We can declare a String type variable and initialize it directly with a group of characters.
ex:- String st = "yellaswamy";
2.We can create a String object using new operator and pass a group of characters to the object.
ex:- String s1 = new String("Ashok");
3.We can create a character array into a string by passing it to the String object.
ex:- char arr[] = {'K','s','w','a','m','y'};
String s2 = new String(arr);
ex:- String s3 = new String(arr,1,4);
o/p: swamy

Ex1:
```
class Test
{
public static void main(String... args)
{
//create a String object or constructing a String
//1.We can declare a String type variable and initialize it directly with a group of characters.
String str1="Yellaswamy";
System.out.println(str1);

//2.We can create a String object using new operator and pass a group of characters to the object.
```

```
String str2=new String("Ashok");
System.out.println(str1);

//3.We can create a character array into a string by passing it to the  String object.

char arr[] = {'K','y','e','l','l','a','s','w','a','m','y'};
                        String s1 = new String(arr);
                        System.out.println(s1);
                String s2 = new String(arr,1,10);
                        System.out.println(s2);
}
}
```

output:

```
D:\Yellaswamy_ClassNotes\Strings>javac Test.java

D:\Yellaswamy_ClassNotes\Strings>java Test
Yellaswamy
Yellaswamy
Kyellaswamy
yellaswamy

D:\Yellaswamy_ClassNotes\Strings>
```

Types of Obejcts:-

There are 2 types.

1.Mutable

2.Immutable

1.Mutable:-

A mutable object is an object where content can be modified.

2.Immutable:-

An immutable object is an object whose content can not be modified.

=>String objects are immutable.its contents cannot be changed.

String class methods:-

String class belongs to java.lang package.

1.String concat(String str):-

concatenates the calling string with str.

note:- '+' will also do the same.

ex:- String s1="hydera"; String s2 = "bad";

String s3 = s1.concat(s2);

o/p: hyderabad

2.int length():-

Returns the length of the string.

String s1 = "Vijayawada";

int n = s1.length();

3.char charAt(int i):-

It extracts only one character from given String.That character at the same ith place.

4.int compareTo(String str):- (case sensitive)

(or)

int compareToIgnoreCase(String str):- (case insensitive)

They are used to compare two strings.

5.boolean equals(String str):- (case sensitive)

(or)

boolean equalsIgnoreCase(String str):- (case insensitive)

It returns true if the calling string equals to str.

6.boolean startsWith(String prefix):-

It returns true if the calling string starts with prefix(begins)

prefix - sub string (or) not.

7.boolean endsWith(String suffix):-

It returns true if the invoking string ends with suffix.

Note:- The above two methods use case sensitive comparison.

8.int indexOf(String str):-

It returns the position number of substring in the main String.So we have to pass substring.

It returns the first occurance of str in the string.

EX:- "This is a book"

int n = str.indexOf("is");

o/p :- n = 2

9.int lastIndexOf(String str):-

It returns last Occurance in the string.

10.String replace(char oldchar,char newchar):-

It returns a new String that is obtained by replacing all characters of 'oldchar' in string with 'newchar'.

11.String substring(int beginIndex):-

It ruturns a new String consisting of all characters from beginindex until the end of the String.

12.String substring(int beginIndex,int endIndex):-

It returns a new String consisting all characters from beginindex until endindex(exclusive).

13.String toLowerCase():-

It converts all characters into lowercase and returns.

14.String toUpperCase():-

IT converts all characters into uppercase and returns.

15.String trim():-

It eleminates all leading and trailing spaces.

StringBuffer Objects are mutable.

Creating StringBuffer objects:-

1.StringBuffer sb = new StringBuffer("hello");

2.StringBuffer sb = new StringBuffer(50);

3.StringBuffer sb = new StringBuffer();

java.lang.StringBuffer methods:-

---

1)StringBuffer append(x):-

x may be int,float,double,char,String (or) StringBuffer.It will be appended to the calling StringBuffer.

2)StringBuffer insert(int offset,x):-

x may be int,float,double,char,String (or) StringBuffer.It will be inserted into the StringBuffer at offset.

3)StringBuffer delete(int start,int end):-

Removes the characters from start to end position.

4)StringBuffer reverse():-

It reverses the all characters in the StringBuffer.

5)String toString():-

Converts the StringBuffer into the String.

purpose:- converts StringBuffer to string class.

6)int length():-

returns the length of the StringBuffer.

7)int indexOf(String str):-

It returns the first position of subString 'str' in the StringBuffer object.

8)int lastIndexOf(String str):-

It returns the last Occurance of substring 'str' in the StringBuffer object.

StringBuilder:-

This class has been added in jdk1.5.0 which has same features like StringBuffer class.These objects are also mutable as are the StringBuffer objects.

IIQ)What is the difference between the StringBuffer and StringBuilder classes?
r) StringBuffer class is synchronized and StringBuilder is not.When the programmer wants to use several threads,he should use StringBuffer as it gives reliable results.If only one thread is used,StringBuilder is prefered,as it improves execution time.

H.W:
1)sort a given group of strings into alphabetical order.
2)find the position of substring in a given main string.
3)Test whether a given string is palindrome or not.

Ex2:

```
D:\Yellaswamy_ClassNotes>javap java.lang.String
Compiled from "String.java"
public final class java.lang.String extends java.lang.Object implements java.io.
Serializable,java.lang.Comparable,java.lang.CharSequence{
   public static final java.util.Comparator CASE_INSENSITIVE_ORDER;
   public java.lang.String();
   public java.lang.String(java.lang.String);
   public java.lang.String(char[]);
   public java.lang.String(char[], int, int);
   public java.lang.String(int[], int, int);
   public java.lang.String(byte[], int, int, int);
   public java.lang.String(byte[], int);
   public java.lang.String(byte[], int, int, java.lang.String)      throws jav
a.io.UnsupportedEncodingException;
   public java.lang.String(byte[], int, int, java.nio.charset.Charset);
   public java.lang.String(byte[], java.lang.String)      throws java.io.Unsup
portedEncodingException;
   public java.lang.String(byte[], java.nio.charset.Charset);
   public java.lang.String(byte[], int, int);
   public java.lang.String(byte[]);
   public java.lang.String(java.lang.StringBuffer);
```

```
    public java.lang.String(java.lang.StringBuilder);
    java.lang.String(int, int, char[]);
    public int length();
    public boolean isEmpty();
    public char charAt(int);
    public int codePointAt(int);
    public int codePointBefore(int);
    public int codePointCount(int, int);
    public int offsetByCodePoints(int, int);
    void getChars(char[], int);
    public void getChars(int, int, char[], int);
    public void getBytes(int, int, byte[], int);
    public byte[] getBytes(java.lang.String)       throws java.io.UnsupportedEnc
odingException;
    public byte[] getBytes(java.nio.charset.Charset);
    public byte[] getBytes();
    public boolean equals(java.lang.Object);
    public boolean contentEquals(java.lang.StringBuffer);
    public boolean contentEquals(java.lang.CharSequence);
    public boolean equalsIgnoreCase(java.lang.String);
    public int compareTo(java.lang.String);
    public int compareToIgnoreCase(java.lang.String);
    public boolean regionMatches(int, java.lang.String, int, int);
    public boolean regionMatches(boolean, int, java.lang.String, int, int);
    public boolean startsWith(java.lang.String, int);
    public boolean startsWith(java.lang.String);
    public boolean endsWith(java.lang.String);
    public int hashCode();
    public int indexOf(int);
    public int indexOf(int, int);
    public int lastIndexOf(int);
    public int lastIndexOf(int, int);
    public int indexOf(java.lang.String);
    public int indexOf(java.lang.String, int);
    static int indexOf(char[], int, int, char[], int, int, int);
    public int lastIndexOf(java.lang.String);
    public int lastIndexOf(java.lang.String, int);
    static int lastIndexOf(char[], int, int, char[], int, int, int);
    public java.lang.String substring(int);
    public java.lang.String substring(int, int);
    public java.lang.CharSequence subSequence(int, int);
    public java.lang.String concat(java.lang.String);
```

```
   public java.lang.String replace(char, char);
   public boolean matches(java.lang.String);
   public boolean contains(java.lang.CharSequence);
   public java.lang.String replaceFirst(java.lang.String, java.lang.String);
   public java.lang.String replaceAll(java.lang.String, java.lang.String);
   public java.lang.String replace(java.lang.CharSequence, java.lang.CharSequen
ce);
   public java.lang.String[] split(java.lang.String, int);
   public java.lang.String[] split(java.lang.String);
   public java.lang.String toLowerCase(java.util.Locale);
   public java.lang.String toLowerCase();
   public java.lang.String toUpperCase(java.util.Locale);
   public java.lang.String toUpperCase();
   public java.lang.String trim();
   public java.lang.String toString();
   public char[] toCharArray();
   public static java.lang.String format(java.lang.String, java.lang.Object[]);

   public static java.lang.String format(java.util.Locale, java.lang.String, ja
va.lang.Object[]);
   public static java.lang.String valueOf(java.lang.Object);
   public static java.lang.String valueOf(char[]);
   public static java.lang.String valueOf(char[], int, int);
   public static java.lang.String copyValueOf(char[], int, int);
   public static java.lang.String copyValueOf(char[]);
   public static java.lang.String valueOf(boolean);
   public static java.lang.String valueOf(char);
   public static java.lang.String valueOf(int);
   public static java.lang.String valueOf(long);
   public static java.lang.String valueOf(float);
   public static java.lang.String valueOf(double);
   public native java.lang.String intern();
   public int compareTo(java.lang.Object);
   static {};
}
```

Programs:

1.

```
import java.io.*;
public class str1
```

```
{
  public static void main(String args[])
  {
      String st1=new String();
      st1="Java";
      byte b[]={65,66,67,68,69,70};
      String st2=new String(b);
      String st3=new String(b,1,3);
      System.out.println(st1);
      System.out.println(st2);
      System.out.println(st3);
   }
}
```

output:

```
D:\Yellaswamy_ClassNotes\Strings>java str1
Java
ABCDEF
BCD
```

Ex2:

```
import java.io.*;
public class str2
{
  public static void main(String args[])
  {
      char ch[]={'a','b','c','d','e','f'};
      String st1=new String("Java");
      StringBuffer sb=new StringBuffer("cmrcet");
      String st2=new String(ch);
      String st3=new String(ch,2,4);
      String st4=new String(sb);
      System.out.println(st1);
      System.out.println(st2);
      System.out.println(st3);
      System.out.println(st4);
   }
}
```

output:

```
D:\Yellaswamy_ClassNotes\Strings>java str2
```

Java
abcdef
cdef
cmrcet

Ex3:

```
import java.io.*;
public class str3
{
  public static void main(String args[])
  {
     String s=new String("welcome");
     int i;
     char ch;
     i=s.length();
     ch=s.charAt(5);
     System.out.println("Length of given string        :  "+i);
     System.out.println("The character at given index   :  " +ch);
  }
}
```

output:

```
D:\Yellaswamy_ClassNotes\Strings>java str3
Length of given string        :  7
The character at given index   :  m
```

Ex4:

```
import java.io.*;
public class str4
{
   public static void main(String args[])
   {
        String s1=new String("java");
        String s2=new String("JAVA");
        String s3=new String("welcome");
        int n;

        n=s1.compareTo(s3);
        if(n==0)
```

```
            System.out.println("s1 and s3 are equal");
        else if(n>0)
            System.out.println("s1 is greater than s3");
        else
            System.out.println("s1 is less than s3");

        n=s1.compareTo(s2);
         if(n==0)
            System.out.println("s1 and s2 are equal");
        else if(n>0)
            System.out.println("s1 is greater than s2");
        else
            System.out.println("s1 is less than s2");

        n=s1.compareToIgnoreCase(s2);
         if(n==0)
            System.out.println("s1 and s2 are equal");
        else if(n>0)
            System.out.println("s1 is greater than s2");
        else
            System.out.println("s1 is less than s2");
    }
}
```

output:

```
D:\Yellaswamy_ClassNotes\Strings>java str4
s1 is less than s3
s1 is greater than s2
s1 and s2 are equal
```

Ex:5

```
import java.io.*;
class str5
{
  public static void main(String args[])
  {
        boolean b;
        StringBuffer sb=new StringBuffer("welcome");
        String s1=new String("welcome");
        String s2=new String(" to cmrcet");
        String s3=new String();
```

```
        s3=s1.concat(s2);
        System.out.println("Concatenated String : "+s3);
        b=s1.contentEquals(sb);
        if(b==true)
            System.out.println("Given 2 strings are equal");
        else
            System.out.println("Given 2 strings are not equal");
    }
}
```

output:

```
D:\Yellaswamy_ClassNotes\Strings>java str5
Concatenated String : welcome to cmrcet
Given 2 strings are equal
```

Ex:6

```
import java.io.*;
class str6
{
  public static void main(String args[])
  {
     char ch[]={'a','b','c','d','e','f'};
     String s1=String.copyValueOf(ch);
     String s2=String.copyValueOf(ch,1,4);
     System.out.println(s1);
     System.out.println(s2);
   }
}
```

output:

```
D:\Yellaswamy_ClassNotes\Strings>java str6
abcdef
bcde
```

Ex:7

```
import java.io.*;
public class str7
{
   public static void main(String args[])
   {
        String s1=new String("WELCOME");
```

```
        String s2=new String("welcome");
        boolean b;

       b=s1.endsWith("come");
        if(b==true)
          System.out.println("true");
        else
          System.out.println("false");

        b=s1.equals(s2);
        if(b==true)
          System.out.println("Given 2 strings are equal");
        else
          System.out.println("Given 2 strings are not equal");

        b=s1.equalsIgnoreCase(s2);
        if(b==true)
          System.out.println("Given 2 strings are equal");
        else
          System.out.println("Given 2 strings are not equal");
    }
}
```

output:

```
D:\Yellaswamy_ClassNotes\Strings>java str7
false
Given 2 strings are not equal
Given 2 strings are equal
```

Ex:8

```
import java.io.*;
public class str8
{
  public static void main(String args[])
  {
        int i;
        String st=new String("WELCOME");
        byte bt[]=new byte[10];
        char ch[]=new char[10];
        bt=st.getBytes();
        System.out.println("Byte Array");
```

```
        for(i=0;i<bt.length;i++)
        {
            System.out.println(bt[i]+"  ");
        }
        st.getChars(3,7,ch,2);
        System.out.println("Character Array");
        for(i=0;i<ch.length;i++)
        {
            System.out.println(ch[i]+"  ");
        }
    }
}
```

output:

```
D:\Yellaswamy_ClassNotes\Strings>java str8
Byte Array
87
69
76
67
79
77
69
Character Array


C
O
M
E
```

Ex:9

```
import java.io.*;
public class str9
{
  public static void main(String args[])
  {
        String s1=new String("welcome");
        String s2=new String("abcdabcdabc");
        int id,hc;
        hc=s1.hashCode();
```

```
        System.out.println("Hash Code of given String : "+hc);
        id=s1.indexOf('e');
        System.out.println("Index : " +id);
        id=s1.indexOf('e',3);
        System.out.println("Index : " +id);
        id=s1.indexOf("come");
        System.out.println("Index : " +id);
        id=s2.indexOf("ab",5);
        System.out.println("Index : " +id);
    }
}
```

output:

```
D:\Yellaswamy_ClassNotes\Strings>java str9
Hash Code of given String : 1233099618
Index : 1
Index : 6
Index : 3
Index : 8
```
