Wrapper classes

Q. What are wrapper classes?

Wrapper class is inbuilt API which help us to perform conversion in JAVA
Means using Wrapper classes we can perform conversion between primitive type to reference type and reference type to primitive type.

There are two types of conversion in JAVA?

1. **Primitive type casting :** primitive type casting means performing conversion between simple data types called primitive type casting.

   There are two types of primitive type casting

   a. **Implicit type casting :** implicit type casting means those conversions performed by the compiler internally called implicit type casting.
   When we have larger data type at left hand side and smaller data type at right hand side then the compiler is able to perform conversion automatically called implicit conversion.

   Example: long a;
   int b=100;
   a=b; // implicit conversion

```
public class ConApp
{ public static void main(String x[])
   {   int a=100;
       long b=a; //implicit conversion
       System.out.printf("B is %d\n",b);
   }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac ConApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java ConApp
B is 100

C:\Program Files\Java\jdk1.8.0_291\bin>
```

   b. **Explicit type casting :** explicit type casting means those conversions performed by developers are called explicit type casting.
   When we have a smaller type of data at left hand side and larger type of data at right hand side then compiler is unable to perform conversion automatically then developer has responsibility to perform conversion manually called as explicit conversion.
   Example:
   int a;
   Long b=100;
   a=(int)b;//explicit conversion

```
public class ConApp
{ public static void main(String x[])
   {   long a=100;
       int b=(int)a; // we conver manually long to int
                     //so it is explicit conversion
       System.out.printf("B is %d\n",b);
   }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac ConApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java ConApp
B is 100

C:\Program Files\Java\jdk1.8.0_291\bin>
```

Note: implicit type casting and explicit type casting get failed when we try to convert referential value to primitive type of value and primitive type of value to referential type of value

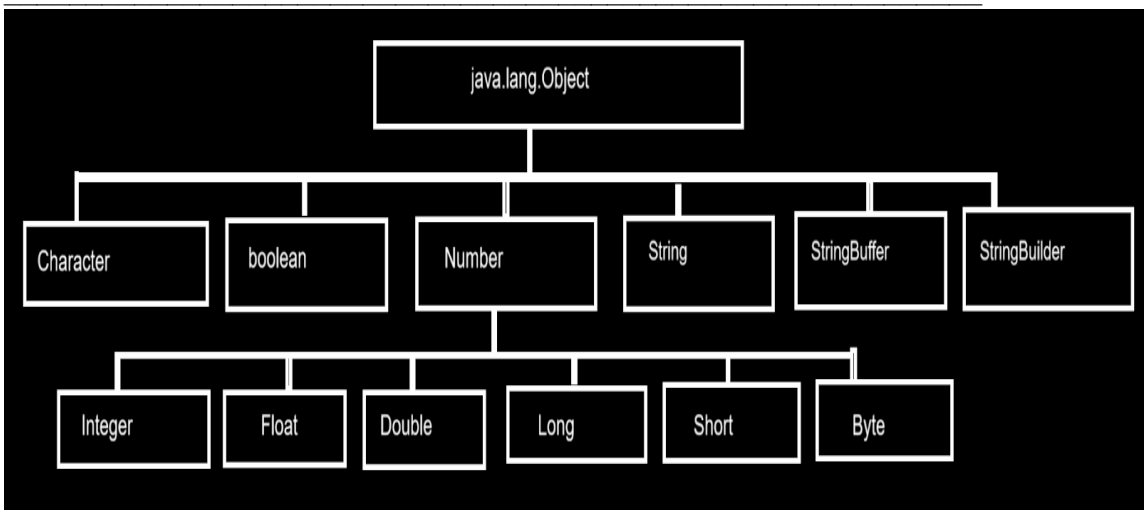
Example:

```
public class ConApp
{ public static void main(String x[])
   {    String str="1234";
        int a;
        a=(int)str;
        System.out.println(a);
   }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac ConApp.java
ConApp.java:5: error: incompatible types: String cannot be converted to int
                a=(int)str;
                       ^
1 error

C:\Program Files\Java\jdk1.8.0_291\bin>
```

Note: if we think about above code we have reference variable String str="1234" and we try to convert this reference variable
to int and we cannot convert reference to primitive or primitive to reference.
If we want to solve this problem JAVA provides some inbuilt classes to us known as Wrapper classes.

Hierarchy of Wrapper classes



```
public class ConApp
{ static Integer a;
  static int b;
  public static void main(String x[])
   {   System.out.println("A is "+a);
       System.out.println("B is "+b);
   }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac ConApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java ConApp
A is null
B is 0

C:\Program Files\Java\jdk1.8.0_291\bin>
```

Note: if we think about above code have statement Integer a; here a is reference variable so the default of a is null and b is primitive type of integer so its default value is 0

There are two types of conversion in Wrapper classes

---

a. **Autoboxing : a**utoboxing means converting primitive value into reference value directly called autoboxing**.**

```
public class AutoBoxApp
{
    public static void main(String x[])
    {
       int a=100;
       Integer b=a;//autoboxing
       System.out.println("B is  "+b);
    }
}
```

Note: if we think about left hand side we code have primitive type of data int a=100 and we have referential type of data name as Integer b and we have statement b=a; here means we store primitive value in to reference value called as autoboxing.

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac AutoBoxApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java AutoBoxApp
B is  100

C:\Program Files\Java\jdk1.8.0_291\bin>
```

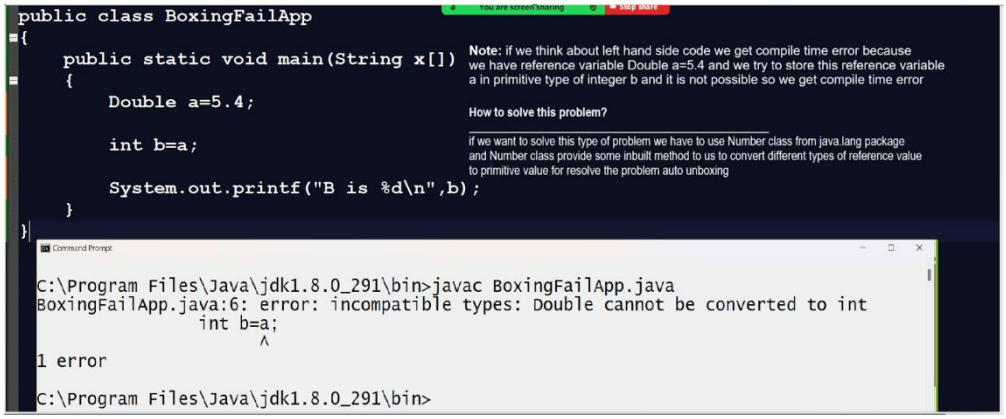
b. **Autounboxing:** autounboxing means converting reference value to primitive value directly called autounboxing.

```
public class AutoUnboxApp
{
    public static void main(String x[])
    {
       Integer a=100;
       int b=a;
       System.out.printf("B is %d\n",b);
    }
}
```

Note: if we think about left hand side we have statement Integer a=100 means here Integer is reference variable and int b=a; here b is primitive type of data and we store reference type of value in primitive type of numeric value called as Auto unboxing.

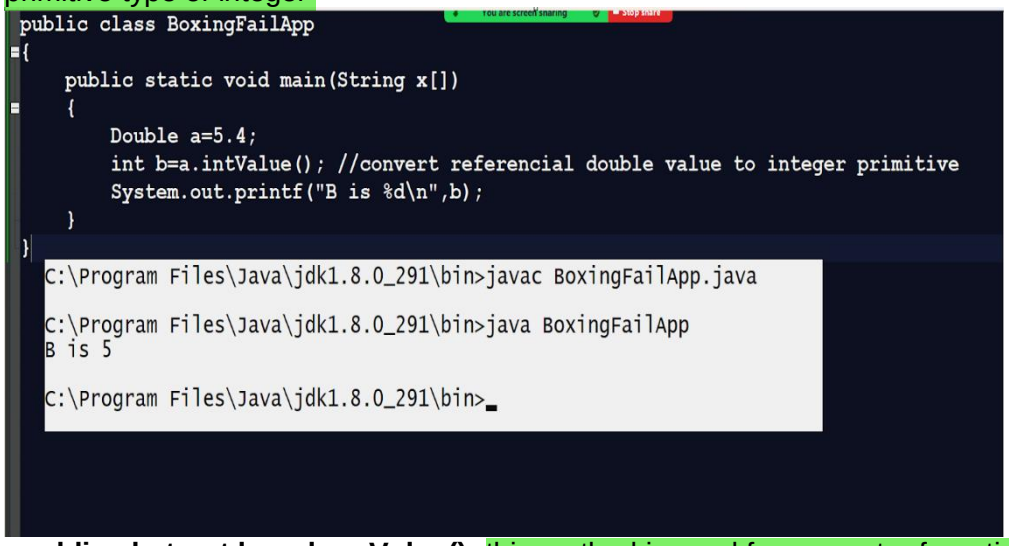
```
C:\Program Files\Java\jdk1.8.0_291\bin>javac AutounboxApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java AutoUnboxApp
B is 100

C:\Program Files\Java\jdk1.8.0_291\bin>
```

Note: if we think about Autoboxing and Autounboxing then it will fail when we try to convert different types of numeric reference to different types of primitive and different types of primitive to different types of reference shown in following code.



Now we want to think about Number class
Number: Number is abstract class from java.lang package and it contains some abstract methods which help us to perform conversion between Numeric type of reference to primitive type value

Methods of Number class
public abstract int intValue(): this method is used for convert referential Numeric value to primitive type of integer



public abstract long longValue(): this method is used for convert referential numeric value to primitive type of long
public abstract float floatValue(): this method is used for convert referential numeric value to primitive type of float
public abstract double doubleValue(): this method is used for convert referential numeric value to double primitive type
public byte byteValue(): this method is used for converting numeric reference value to primitive byte type.
public short shortValue(): this method is used for converting the numeric reference value to primitive short type.

```
public class BoxingFailApp
{
    public static void main(String x[])
    {
        Double a=5.4;
        long l=a.longValue();//convert referencial double to lont value
        System.out.println("Long l "+l);
        Float f=5.6f;
        int d=f.intValue();//convert float to int
        System.out.println("Int D is "+d);
    }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac BoxingFailApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java BoxingFailApp
Long l 5
Int D is 5

C:\Program Files\Java\jdk1.8.0_291\bin>
```

parseXXX(): this is the method present in every wrapper class and it is a static method present in every wrapper class and the goal of this method is convert string value to primitive type of value.

Syntax: int var=Integer.parseInt(String): convert string to integer
float var=Float.parseFloat(String): convert string to float
double var=Double.parseDouble(String): convert string to double
long var=Long.parseLong(String): convert string to long type
short var=Short.parseShort(String): convert string to short

Etc

Note: this method may generate NumberFormatException to us at program run time. Because if we think about string then it may be possible it contains some non numeric values like space or any special character or alphabet then there is possible we get errors at program run time or NumberFormatException at program run time.

```
public class StringToNum
{
    public static void main(String x[])
    {
        String s="1234"; //1234 is numeric but in string format
        int b=Integer.parseInt(s); //we convert string value to integer
        System.out.println("B is  "+b);
    }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac StringToNum.java

C:\Program Files\Java\jdk1.8.0_291\bin>java StringToNum
B is  1234

C:\Program Files\Java\jdk1.8.0_291\bin>
```

Note: possibility exception in given code

```
public class StringToNum
{
    public static void main(String x[])
    {
        String s="1234 "; //1234 is numeric but in string format
        int b=Integer.parseInt(s); //we convert string value to integer
        System.out.println("B is  "+b);
    }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac StringToNum.java

C:\Program Files\Java\jdk1.8.0_291\bin>java StringToNum
Exception in thread "main" java.lang.NumberFormatException: For input string: "1234 "
        at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
        at java.lang.Integer.parseInt(Integer.java:580)
        at java.lang.Integer.parseInt(Integer.java:615)
        at StringToNum.main(StringToNum.java:6)

C:\Program Files\Java\jdk1.8.0_291\bin>
```

Note: we get error because we have String String s="1234 " here 1234 is integer value so we can easily convert integer but we have space present in String and we cannot convert space to integer so we get NumberFormatException at program run time so we can handle it using exception handling shown in following code.

Example with source code

```
public class StringToNum
{   public static void main(String x[])
    { try{
        String s="1234 "; //1234 is numeric but in string format
        int b=Integer.parseInt(s); //we convert string value to integer
        System.out.println("B is  "+b);
      }
      catch(NumberFormatException ex)
      { System.out.println("error is "+ex);
      }
    }
}
```

```
Select Command Prompt

C:\Program Files\Java\jdk1.8.0_291\bin>javac StringToNum.java

C:\Program Files\Java\jdk1.8.0_291\bin>java StringToNum
error is java.lang.NumberFormatException: For input string: "1234 "

C:\Program Files\Java\jdk1.8.0_291\bin>_
```

valueOf(): this method is used for converting any primitive value to reference variable means convert primitive type of numeric value to reference type or convert numeric value to string type of value also.

```
public class PrimToRef
{   public static void main(String x[])
    {  int a=1234; // this is primitive type of vlaue
       String s=String.valueOf(a); // we convert primitive to reference
       System.out.println(s);
       Float b=Float.valueOf(a);//convert primitive integer to float reference
       System.out.println(b);
    }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac PrimToRef.java

C:\Program Files\Java\jdk1.8.0_291\bin>java PrimToRef
1234
1234.0

C:\Program Files\Java\jdk1.8.0_291\bin>_
```

String toString(): convert any referential value to string type.

```
public class PrimToRef
{   public static void main(String x[])
    {
        Integer a=5;
        String s=a.toString();//we convert Referencial integer to string
        System.out.println(s);
    }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac PrimToRef.java

C:\Program Files\Java\jdk1.8.0_291\bin>java PrimToRef
5

C:\Program Files\Java\jdk1.8.0_291\bin>_
```

String Handling in JAVA?

Q. What is String in JAVA?

String is immutable objects in JAVA means once we initialize value then we cannot modify its later called as immutable

Note: in Java Every “” (double quote) is considered a string object.

If you want to create a string in java we have two ways.

a. Using initialization technique
Syntax: String variable=”value”;
Example: String s=”good”;
b. Using new keyword

Syntax: String variable=new String(“value”);

Example: String s = new String(“good”);

```
public class StringApp
{
    public static void main(String x[])
    {
       String s="good";
       System.out.println("String is  "+s);
       String s1 = new String("good");
       System.out.println("String s1 "+s1);
    }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac StringApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java StringApp
String is  good
String s1 good

C:\Program Files\Java\jdk1.8.0_291\bin>
```

Q. What is the difference between string creation using initialization and using a new keyword?

Using initialization approach: when we create a string using initialization technique then the string is created in string constant pool and when we have the same string value then internally JVM does not create two different string objects in memory just create a single string object and share its reference in different variables or address in different reference variable.

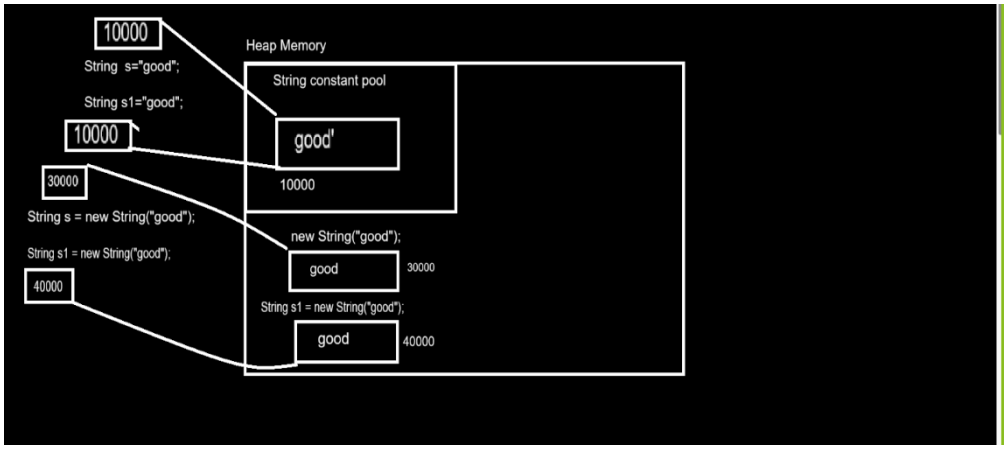
Using a new keyword approach: when we create a string using a new keyword then JVM creates a string object in the heap section of memory and when we have string with the same value or different value then JVM creates a new string object every time.

Q. What is the heap section?

Heap section is part of JVM memory and which is responsible for allocate memory at run time as well as destroy the memory at run time using garbage collection technique means Heap section can allocate memory of object or array at run time by using new keyword and it is part of JVM memory

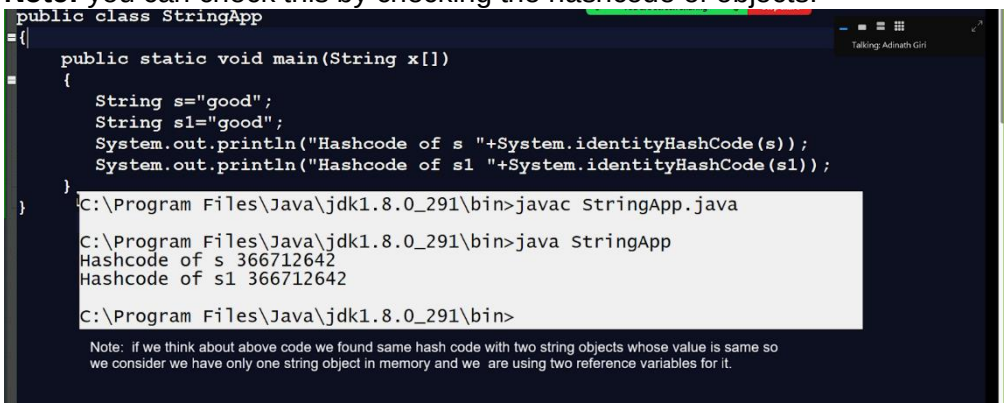
Q. What is a string constant pool?

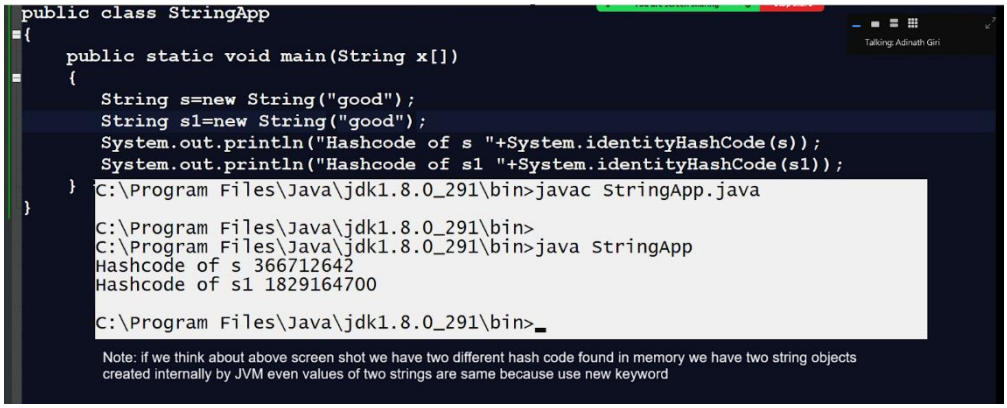
The Java String constant pool is an area in heap memory where Java stores string literals and when the same value string literals is found then not allowed to create a new string object just share its reference in different string variables.



How can we prove the above diagram?

Note: you can check this by checking the hashcode of objects.





If we want to create a String object in java we have the following constructors.

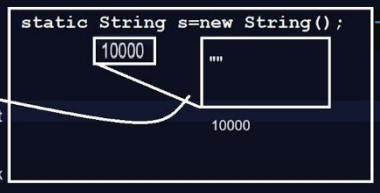
String(): this will create an empty string object in the heap section of memory.

```
public class StringApp
{    static String s;
     public static void main(String x[])
     {
        System.out.println(s.length());
     }
}
```

Note: if we think about left hand side code we have String s; here s is our reference and we not allocate memory for that so the default of s is null and if we think about main method we use s.length() means we call function of class without its memory and it is possible so we get NullPointerException to us.

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac StringApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java StringApp
Exception in thread "main" java.lang.NullPointerException
        at StringApp.main(StringApp.java:5)
```

```
public class StringApp
{    static String s=new String();
     public static void main(String x[])
     {
        System.out.println(s.length());
     }
}
```

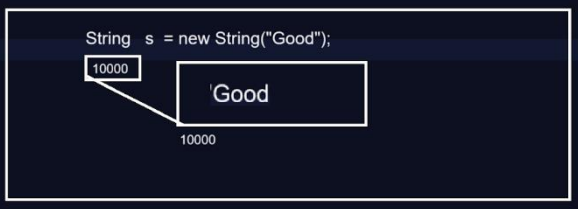


Note: if we think about above output we have statement String s = new String(); here s is reference of String class and we allocate memory for string and store its address in reference variable s and string object has blank value so the we get length is 0.

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac StringApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java StringApp
0

C:\Program Files\Java\jdk1.8.0_291\bin>
```

String(String s) : this constructor can create string objects with some initialised value in memory.

```
public class StringApp
{
     public static void main(String x[])
     {
         String s =  new String("Good");
         System.out.println(s);
     }
}
```



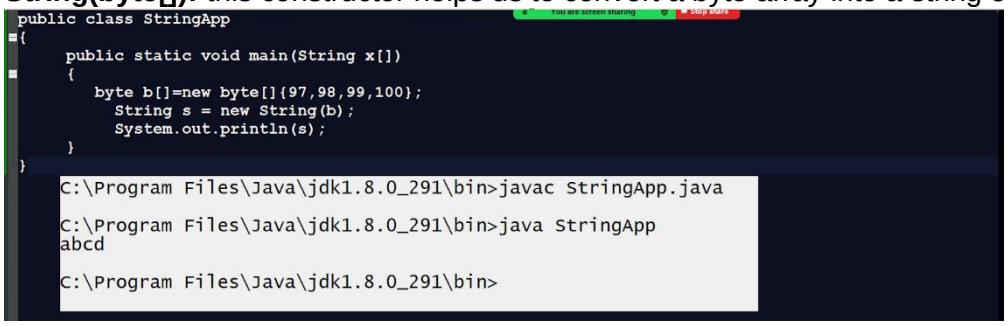
String(char[]): this constructor helps us to create a string by using character array to convert a character array into string object.

```
public class StringApp
{
     public static void main(String x[])
     {
          char ch[]=new char[]{'a','b','c','d'};
          String s = new String(ch);
          System.out.println(s);
     }
}
```

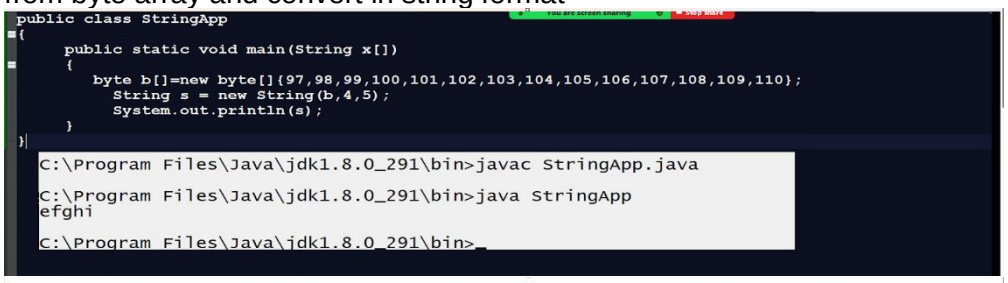
String(char[],int offset,int length): this constructor helps us to convert a specified portion of the character array into string format.
char []: this parameter accept character array as parameter
int offset: this parameter help us to set index from we want to extract data

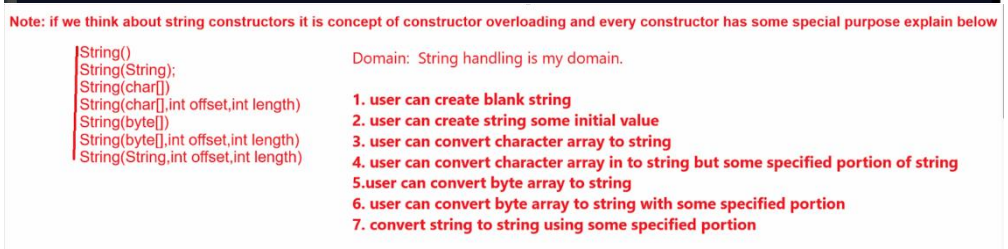
int length: this parameter helps us to decide the length of data which we want to extract.

String(byte[]): this constructor helps us to convert a byte array into a string object.



String(byte[],int offset,int length): this constructor help us to extract some specified portion from byte array and convert in string format





Important points related with string

---

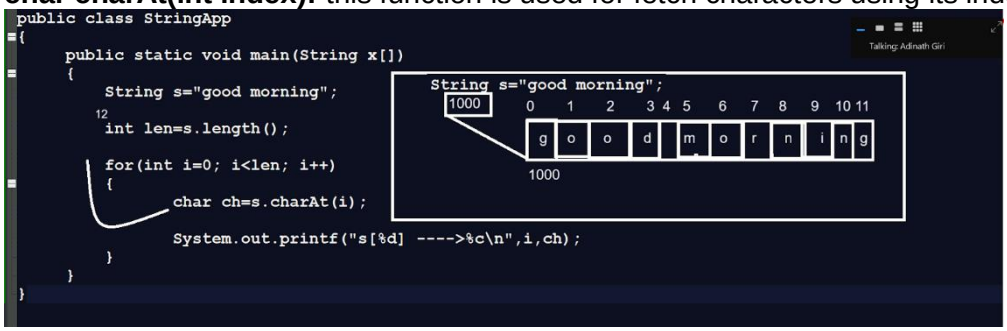
1. **String is thread safe :** means multiple threads cannot access String object simultaneously if one thread object uses string object then another thread cannot access within that period can access when previous thread finishes.
   Because string class is final class so it is thread safe.

2. String class cannot inherit in another class because internally it is final class and final class cannot inherit in another class.
3. String class is immutable class means once we initialise some value in string object we cannot modify it later.
4. Not need to import package to use string class because it is member of java.lang and java.lang is default package so we not need to import

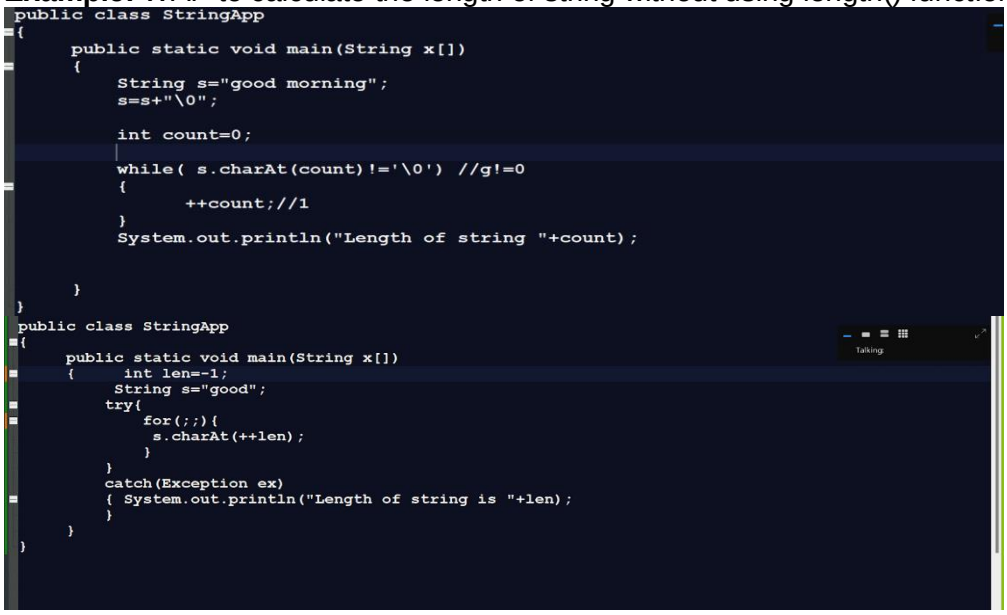
Methods of string class

---

int length(): this function helps us to calculate length of string.
char charAt(int index): this function is used for fetch characters using its index.

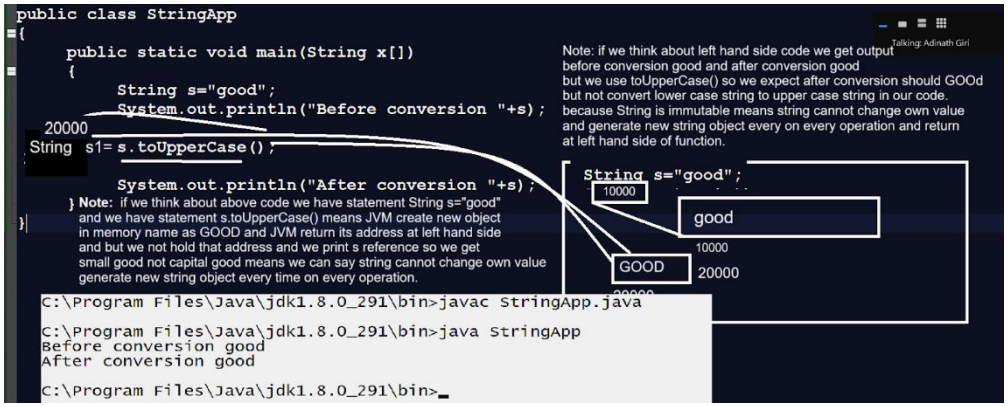


Example: WAP to calculate the length of string without using length() function



String toUpperCase(): this function can convert lower case string to upper case string.

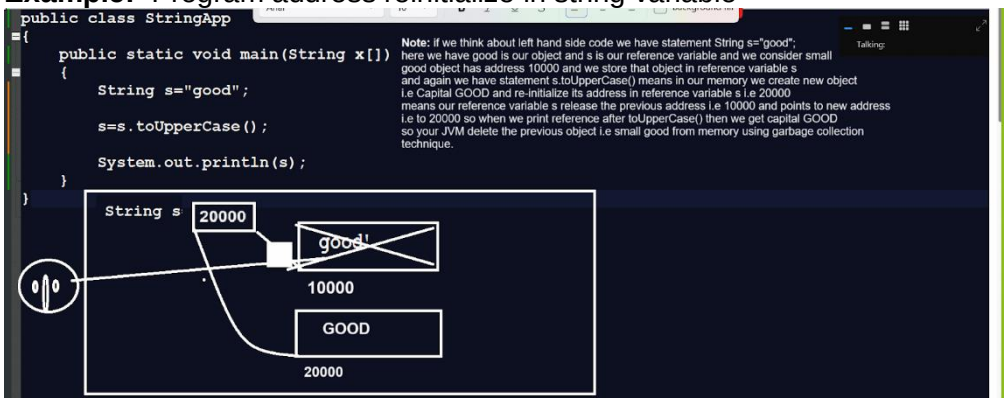
We solve above problem in next diagram



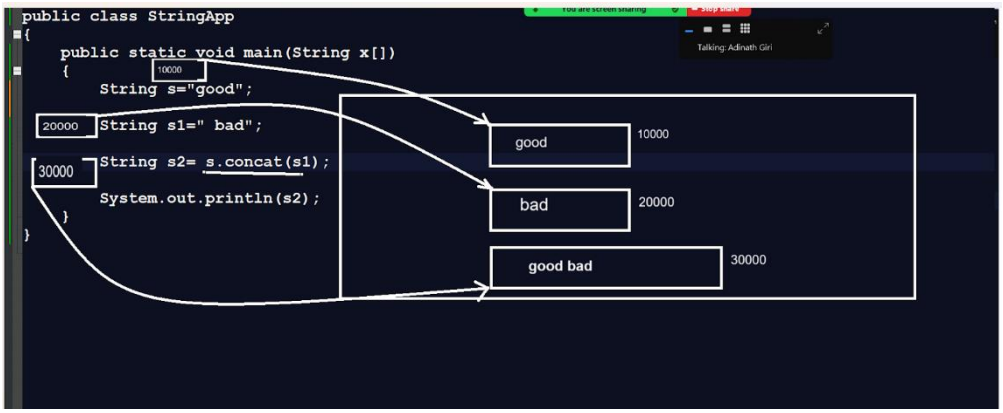
Example with source code

```
public class StringApp
{
    public static void main(String x[])
          {       String s="good";
System.out.println("Before conversion "+s+"\t"+System.identityHashCode(s));
String s1=s.toUpperCase();
 System.out.println("After conversion "+s1+"\t"+System.identityHashCode(s1));
          }
}
```

Example: Program address reinitialize in string variable

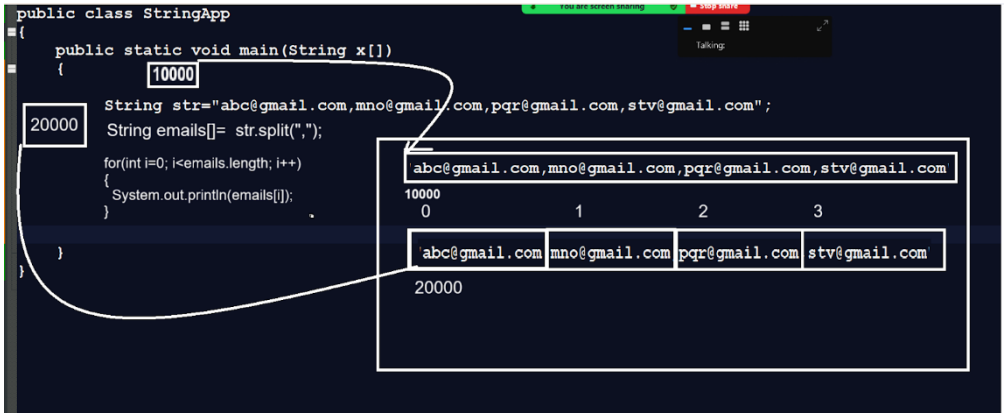


String concat(String): this method is used for combining the two strings and returning the third string as a new resultant at the left hand side means combining two string objects and generating the new third string object.



String trim(): this method is used for removing the white spaces at beginning and ending of string.

String [] split(String): this method can split the string using some specified character and return data in the form of an array.



int indexOf(String): this method is used for search string or specified string data from string and return its index if string data found otherwise return -1

```
public class StringApp
{
    public static void main(String x[])
    {
       String str="good morning india";
       int index= str.indexOf("morning");
       if(index!=-1)//5!=-1
       { System.out.println("Data found");
       }
       else
       { System.out.println("Data not found");
       }
    }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac StringApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java StringApp
Data found

C:\Program Files\Java\jdk1.8.0_291\bin>
```

String subString(int startIndex,int endIndex): this method is used for extracting the some specified port of string between two indexes and storing its data at the left hand side.

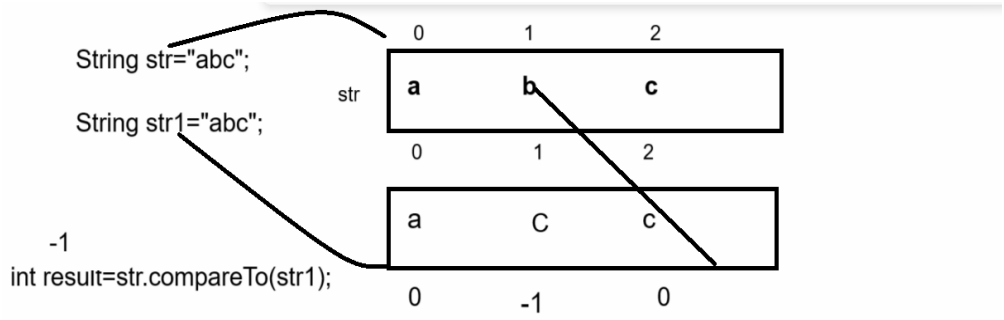
```
public class StringApp
{
    public static void main(String x[])
    {
       String str="good morning india";

       String extractedData=str.substring(5,12);

       System.out.println(extractedData);
    }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac StringApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java StringApp
morning

C:\Program Files\Java\jdk1.8.0_291\bin>
```

```
public class StringApp
{
    public static void main(String x[])
    {
       String str="good morning india";

       String extractedData=str.substring(5,12);

       System.out.println(extractedData);
    }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac StringApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java StringApp
morning

C:\Program Files\Java\jdk1.8.0_291\bin>
```

int compareTo(String): this method can compare two strings with each other by using lexical order means by using alphabetic comparison means compare alphabet of every index same alphabet found in two string return 0 if different alphabet found return first mismatch ascii code difference




```
public class StringApp
{
    public static void main(String x[])
    {
       String s="abc";
       String s1="abc";

       int value= s.compareTo(s1);
        if(value==0)
        { System.out.println("Strings are equal");
        }
        else
        { System.out.println("Strings are not equal");
        }
    }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac StringApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java StringApp
Strings are equal

C:\Program Files\Java\jdk1.8.0_291\bin>
```

boolean equals(Object) : this method can compare any kind of object using its internal content and satisfy java object comparison rules and return true if objects are equal otherwise false.

```
public class StringApp
{
    public static void main(String x[])
    {
       String s="abc";
       String s1="abc";
       boolean result= s.equals(s1);
        if(result)
        { System.out.println("Strings are equal");
        }
        else
        { System.out.println("Strings are not equal");
        }
    }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac StringApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java StringApp
Strings are equal
```

Q. What is the difference between compareTo() and equals()?

a. compareTo() method can compare only strings objects not other and equals() method can compare any kind of object in java

b. compareTo() method is originally member of string class and equals() method of java.lang.Object and Object is parent of every class in JAVA

c. Return type of compareTo() method is int and return type of equals() method is boolean

d. compareTo() compares the string by using lexicol order and equals() method can compare string by using its hashcode.

int compareToIgnoreCase(String): this method help us to compare two strings without checking its internal case

```
public class StringApp
{
    public static void main(String x[])
    {
       String s="abc";
       String s1="ABC";
       int result=s.compareToIgnoreCase(s1);
        if(result==0)
        { System.out.println("Strings are equal");
        }
        else
        { System.out.println("Strings are not equal");
        }
    }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac StringApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java StringApp
Strings are equal

C:\Program Files\Java\jdk1.8.0_291\bin>
```

Example of lastIndexOf() method?

```
public class StringApp
{
    public static void main(String x[])
    {
        String str="good morning india good afternoon india in 2025";

        int index=str.lastIndexOf("india");
        System.out.println(index);
    }
}
```

Note: lastIndexOf() method is used for search index of last occurred data means when we data in string but data may be present repetitively then lastIndexOf() method return index of last occurred data.

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac StringApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java StringApp
34
```

boolean endsWith(String): this method is used to check if a string ends with specified data or not if string ends with specified data return true otherwise return false.

```
public class StringApp
{
    public static void main(String x[])
    {
       String str="Ganesh";
       boolean result=str.endsWith("sh");
       if(result)
       { System.out.println("String ends with sh");
       }
       else
       { System.out.println("Strings not ends with sh");
       }
    }
}
```

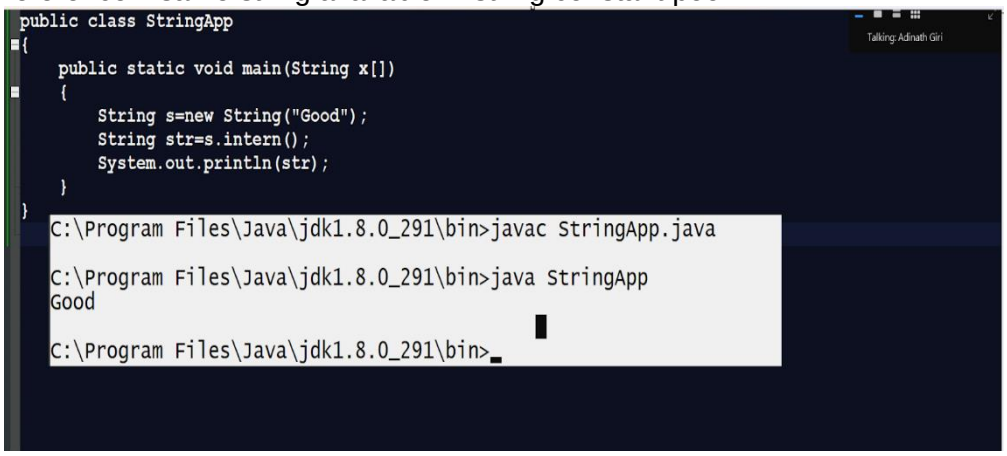
```
C:\Program Files\Java\jdk1.8.0_291\bin>javac StringApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java StringApp
String ends with sh

C:\Program Files\Java\jdk1.8.0_291\bin>
```

boolean startsWith(String): this method checks if a string starts with specified character or not if start with specified character return true otherwise return false.

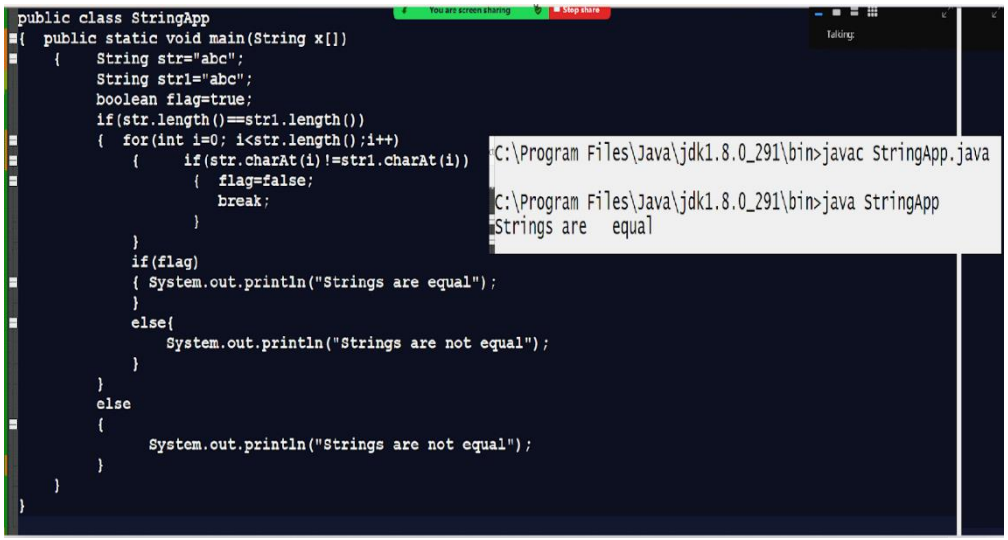
```
public class StringApp
{
    public static void main(String x[])
    {
       String str="Ganesh";
       boolean result=str.startsWith("G");
       if(result)
       { System.out.println("String ends with G");
       }
       else
       { System.out.println("Strings not ends with sh");
       }
    }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac StringApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java StringApp
String ends with G

C:\Program Files\Java\jdk1.8.0_291\bin>
```

String intern(): this method can shift string in string constant pool from heap and also return reference if same string available in string constant pool.



Example: WAP to compare two strings in java without using int compareTo()?



Example: WAP to extract a digit from string and calculate its sum?

Input: abc123mno456stv

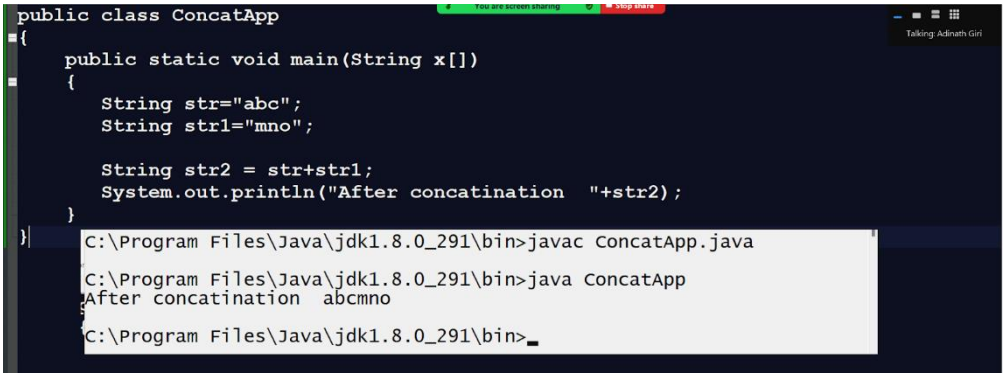
```
public class StringApp
{  public static void main(String x[])
        {    String str="abc1234mno567stv";
             int sum=0;

                  for(int i=0; i<str.length();i++)
                  {
                           char ch=str.charAt(i);
                           if(ch>=48 && ch<=57)
                           {   sum=sum+((int)ch-48);
                                    System.out.printf("%d\t",((int)ch-48));
                           }
                  }
```

```
            System.out.printf("\nSum of all values is %d\n",sum);

        }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac StringApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java StringApp
1       2       3       4       5       6       7
Sum of all values is 28

C:\Program Files\Java\jdk1.8.0_291\bin>
```

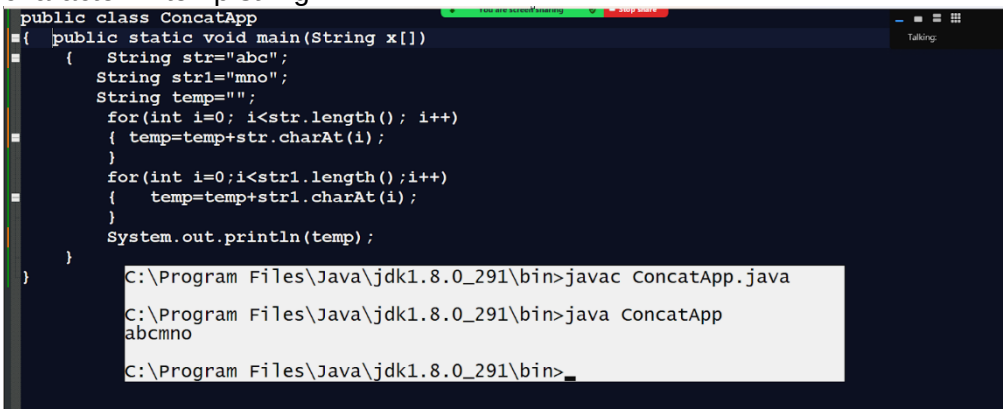
Example: WAP to concat two string without using concat() function

a. **First approach using + (sign) :** In Java we can combine two strings by using + operator



Note: we can convert any value or concat any value with string value.

b. We can take temporary string and initialize it using "" and travel the first string and concat every character in temp string and travel secon string and concat every character in temp string.



WAP to check if strings end with specified data or not without using endsWith()?

```
import java.util.*;
public class CheckEndsWithApp
{  public static void main(String x[])
        {
```

```
        Scanner xyz  = new Scanner(System.in);
            System.out.println("enter original string");
            String str=xyz.nextLine();
            System.out.println("Check strings ends with");
            String str1=xyz.nextLine();
            boolean b= myEndsWith(str,str1);
            if(b){
                    System.out.println("Strings ends with  "+str1);
            }
            else
            {  System.out.println("Strings not ends with  "+str1);
            }
    }
    public static boolean myEndsWith(String original,String end)
    {  int index=original.lastIndexOf(end);
            if(index!=-1)
            { return true;
            }
            else{
              return false;
            }
    }
}
```

StringBuffer and StringBuilder class

StringBuffer and StringBuilder are the mutable classes of JAVA
Mutable means those values can change later or after initialization called mutable classes.

StringBuffer class

Important points related with StringBuffer

a. StringBuffer is mutable class of java

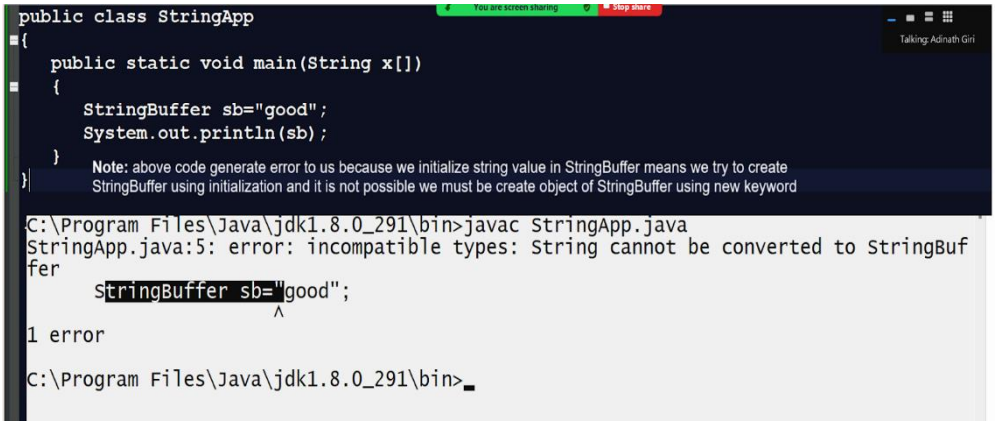
b. StringBuffer can create only its object in heap section means we can create object of StringBuffer by using only new keywords not using initialization

c. StringBuffer is a thread safe class that means an object of StringBuffer can be used by only one thread at time means StringBuffer object cannot be used by more than one thread simultaneously as well as StringBuffer methods are synchronized.

d. StringBuffer methods are synchronized so it is thread safe

e. StringBuffer have some additional methods as compare with string like as append(),insert(),delete() etc

Example of StringBuffer



Example of StringBuffer using new keyword

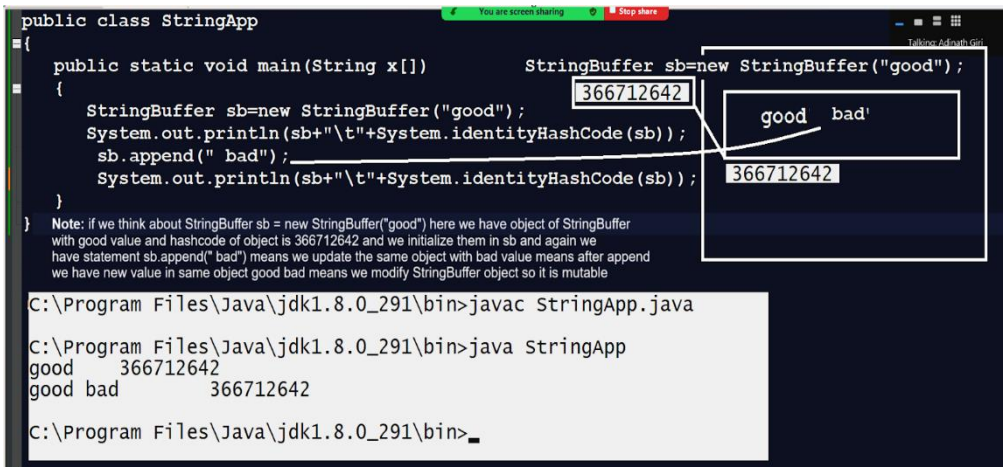
Additional methods of StringBuffer and StringBuilder

---

void append(int), void append(float),void append(double),void append(long),void append(short) ,void append(Object) etc

---

This is the overloaded method with the StringBuffer object means using this method we can append any kind of data at the end of the StringBuffer object and not generate new object after append like as string modify existing object data or content



void insert(int index, int data), void insert(int index,float data), void insert(int index,double data) etc

This is also overloaded method with all data types in JAVA and using this method we can insert data on specified index in StringBuffer or StringBuilder object and shift previous data of that object from specified index in forward direction

```
public class StringApp
{ public static void main(String x[])
   {
      StringBuffer sb=new StringBuffer("good morning india");
      System.out.println(sb+"\t"+System.identityHashCode(sb));
       sb.insert(5," Afternoon & ");
       System.out.println(sb+"\t"+System.identityHashCode(sb));
   }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac StringApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java StringApp
good morning india      366712642
good  Afternoon & morning india 366712642

C:\Program Files\Java\jdk1.8.0_291\bin>
```

void delete(int startIndex,int endIndex): this method is used for delete the data between two specified indexes and update StringBuffer or StringBuilder object

```
public class StringApp
{ public static void main(String x[])
   {
      StringBuffer sb=new StringBuffer("good morning india");
      System.out.println(sb+"\t"+System.identityHashCode(sb));
         sb.delete(5,12);
       System.out.println(sb+"\t"+System.identityHashCode(sb));
   }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac StringApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java StringApp
good morning india      366712642
good  india     366712642

C:\Program Files\Java\jdk1.8.0_291\bin>
```

void reverse(): this method is used for reverse the string value.

```
public class StringApp
{ public static void main(String x[])
   {
      StringBuffer sb=new StringBuffer("good morning india");
      System.out.println(sb+"\t"+System.identityHashCode(sb));
         sb.reverse();
       System.out.println(sb+"\t"+System.identityHashCode(sb));
   }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac StringApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java StringApp
good morning india      366712642
aidni gninrom doog      366712642

C:\Program Files\Java\jdk1.8.0_291\bin>
```

void replace(String oldstring,String newString): this method can replace the string object from old string to new string.

```
public class StringApp
{ public static void main(String x[])
   {
      StringBuffer sb=new StringBuffer("good morning india");
      System.out.println(sb+"\t"+System.identityHashCode(sb));
         sb.replace(5,12,"afternoon");
       System.out.println(sb+"\t"+System.identityHashCode(sb));
   }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>java StringApp
good morning india      366712642
good afternoon india    366712642

C:\Program Files\Java\jdk1.8.0_291\bin>
```

```
public class StringApp
{ public static void main(String x[])
   {
      StringBuilder sb=new StringBuilder("good morning india");
      System.out.println(sb+"\t"+System.identityHashCode(sb));
         sb.replace(5,12,"afternoon");
       System.out.println(sb+"\t"+System.identityHashCode(sb));
   }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac StringApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java StringApp
good morning india      366712642
good afternoon india    366712642

C:\Program Files\Java\jdk1.8.0_291\bin>
```

Q. What is the difference between StringBuffer and StringBuilder?

a. StringBuffer object is thread safe and StringBuilder object is not thread safe
b. StringBuffer methods are synchronized and StringBuider methods are not synchronized.
c. StringBuffer is slower than StringBuilder
d. StringBuffer recommend when we want to thread safety but StringBuilder recommend in Asynchronous environment
e. StringBuffer present in JAVA from JDK 1.0 version and StringBuilder introduced in JDK 1.5 version of JAVA

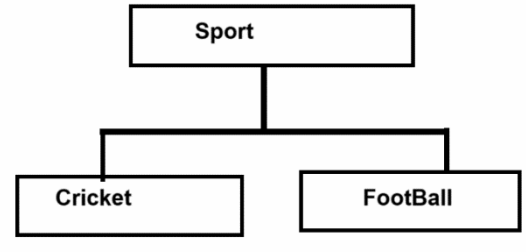
Q. What is the difference between String and StringBuilder?

a. String is immutable class of Java and StringBuilder is mutable class of JAVA
b. String can create using initialization or using new keyword means can create using string constant pool or using heap but StringBuilder object must be create using new keyword means always create heap section of memory
c. String is thread safe object and StringBuilder is not thread safe
d. StringBuilder has some additional method as compare with String like as append(),insert(),delete() etc

2. Referential type casting

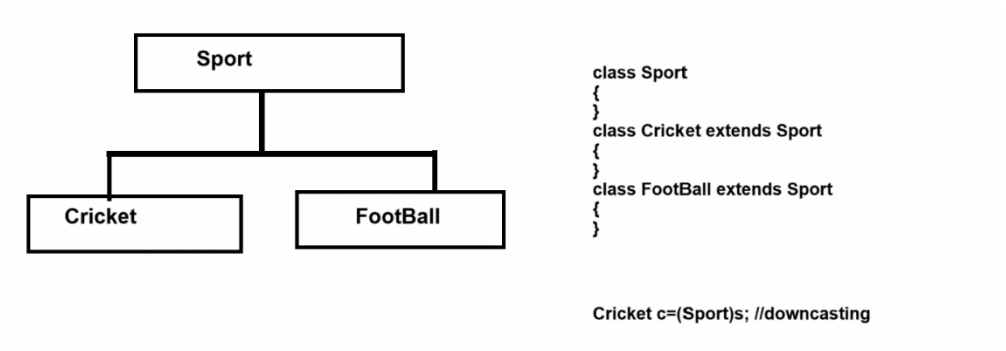
There are two types of referential type casting

a. Upcasting : when we convert a child object into parent object called as upcasting and normally in upcasting we create reference of parent and object of child class.




```
class Sport
{
}
class Cricket extends Sport
{
}
class FootBall extends Sport
{
}

Sport s = new Cricket(); //upcasting
```

b. **Downcasting**: Downcasting means when we convert parent object in to child object called as downcasting



Assignments

Interview Question

Q1. What are wrapper classes and why use them?
Q2. Explain types of type casting with an example?
Q3. What is autoboxing and autounboxing?
Q4. Explain Number class with its all methods and source code examples?
Q5. Explain the valueOf() method with an example?
Q6. Explain hierarchy of Wrapper classes?

Program

Q1. WAP to input string and extract digit from it and calculate its addition?
Input: abc123mno456pqr
Output: 1+2+3+4+5+6 = 21

Q2. WAP to input string and calculate the vowel,consonant and special symbols in string?

Q3. WAP to input string and reverse every word of string without using inbuilt methods?
Input: good morning india
Output: doog gninrom aidni

Q4. WAP to input string and convert every word first letter in capital?
Input: good morning india.
Output: Good Morning India.

Date: 24/02/2025

Interview Question

Q1. What is StringBuffer and StringBuilder class?
Q2. What is the difference between StringBuffer and StringBuilder class?
Q3. What is the difference between String and StringBuilder class?
Q4. What are similarities between String and StringBuffer?

Q5. What is string constant pool in JAVA?
Q6. Can we create a StringBuffer or Builder using the initialization technique?
Q7. Explain 10 methods of String with examples and memory diagrams?
Q8. Explain the 5 methods of StringBuffer and StringBuilder class?

Program for practice?

Q1. Java program to print Even length words in a String?

Input: s = "This is a java language"

Output: This is java language

Explanation: All the elements with the length even are printed.
"This" length is 4 so printed whereas "a" length is 1 so not Printed.

Q2. Java String Program to Insert a String into Another String

Input: originalString = "JavaLanguage",
stringToBeInserted = "is",
index = 4

Output: "JavaisLanguage"

Explanation: Adding the new String to orignal String at the index given.

Q3. Java String program to check whether a string is a Palindrome

Input: str = “aba”

Output: yes

Explanation: Palindrome is String which can be read same both forth and reverse side or can be said String whose orignal string is same as reverse of String.
"AbbA" , "DaD" , etc these are some examples of Palindrom String.

Q4.Java String Program to Check Anagram

Input: str1 = "Listen"
str2 = "Silent"

Output: Yes

Explanation: A string is called Anagram of other string when the it contains the same characters, only the order of characters can be different.
Example Listen -> E:1 , I:1 , L: 1 , N:1 , S:1 , T:1
Silent -> E:1 , I:1 , L: 1 , N:1 , S:1 , T:1

As the occurence of elements are same in both the String hence they are anagram of each other.

Q5.Java String Program to Reverse a String

Input: "abcd"

Output: dcba

Q6. Java String Program to Swapping Pair of Characters

Input: str = “GIRITECHHUB”

Output: IGIRETHCUHB

Q7. Java String Program to Replace a Character at a Specific Index

Input: str = "JAVA IS FOOD Programming " , index = 8 , ch = 'G'

Output: "JAVA IS GOOD Programming"

Q8. Java String Program to Remove Leading Zeros

Input: 000012356098

Output: 12356098

Explanation: Removing all the elements from the beginning of String which doesn't add any value to the number.

Q9.Java String Program to Sort a String

Q10. Java String Program to Compare Two Strings

Q11.Program to Find the Sum of Two Large Numbers.
Input : str1 = "7777555511111111",
str2 = "3332222221111"

Output : 7780887733332222

Q12.
Program to Extract Substring from a String with Equal 0, 1, and 2.
Input: str = “102100211”

Output: 5

Explanation: "102" , "021" , "210" , " 021" , "210021" these combinations can be formed where the occurrences of 0 , 1 and 2 all are equal.

Date : 25/02/2024

Q1. Explain JVM Memory diagram?

Q2. What is the permanent generation section and why use it?
Q3. What is the heap section and why use it?
Q4. Explain types of heap and its uses?
Q5. What is garbage collection and explain its type?
Q6. What is Major and Minor GC?
Q7. What is Stack and why use it?
Q8. What is the difference between stack and heap?
Q9. What is a runtime constant pool and why use it?
Q10. What is a memory constant pool and where is it present and why use it?

Date: 26/2/2025

Q1. What is thread in JAVA and why use it?
Q2. What is the process ?
Q3. What is the difference between thread and process?
Q4. What is multithreading in JAVA?
Q5. How many ways to implement the thread in JAVA?
Q6. How to implement the thread using Thread class Explain with an example?
Q7. What is a runnable interface in JAVA?
Q8. is it true that run() is a method of Thread class?
Q9. Explain 6 methods of Thread class with an example?
Q10. is it true that run() needs to be called manually by the developer?

Program for practice

Q.Write a menu-driven java program to enter the string and following oprations.

input String :I am indian.

first ouput : naidni ma I

Secound output : I ma naidni

Third outout : indian am I

Fourth output : I am iNdiaN

Fifth Output : I aM indiaN