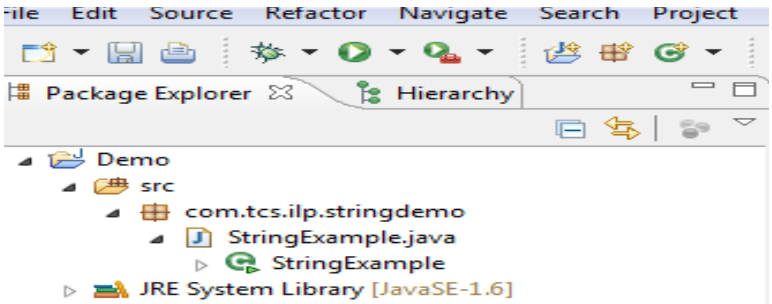
1.Example to show basic String operations.



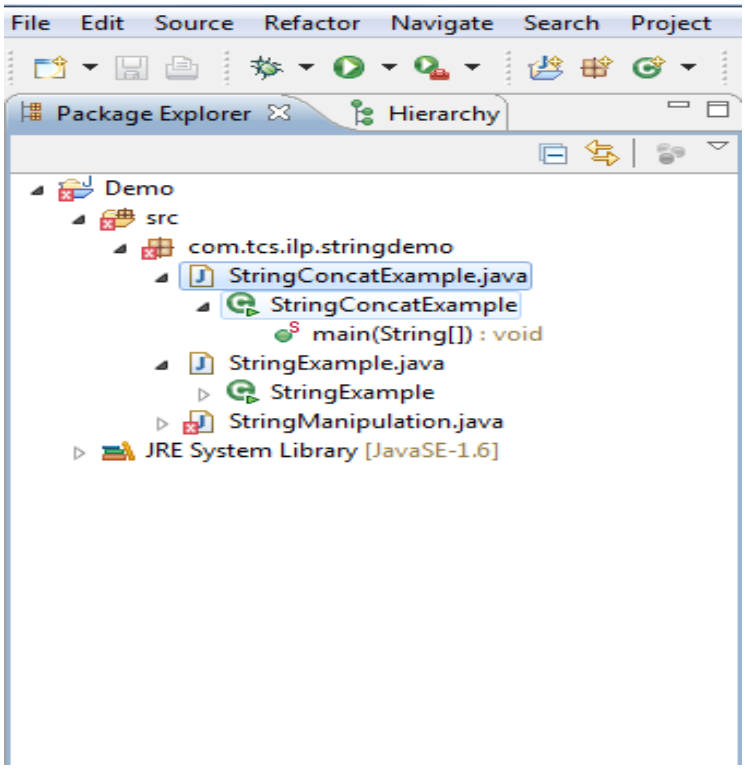

```
/*This program provides some example Java code of how to work with the String
class.
It shows the use of String Literals, manipulation of Strings, as well as converting
them to numbers and back again.*/

package com.tcs.ilp.stringdemo;

public class StringExample {

          public static void main(String[] args) {

              //Use a string literal to assign a value to the String
              String address = "I live at 22b Baker Street!";

              System.out.println("Here is Sherlock's address: " + address);


              //A char array can be used to make a String
              char characterArray[] = {'C', 'h', 'a','r','a','c','t','e','r','s'};
              String characterString = new String(characterArray);

              //or even a byte arrray
              byte byteArray[] = {67,104,97,114,97,99,116,101,114,115};
              String byteString = new String(byteArray);

              System.out.println("Char Array: " + characterString);
              System.out.println("Byte Array: " + byteString);
```

```
                //Dealing with unusual characters by using the Unicode value
                String footballPlayer = "Thomas M\u00FCller plays for Germany.";
                System.out.println(footballPlayer);

                //Escape sequences for characters
                String speech = "\"I say old chap\", he said to me";
                String backSlashNewLine = "The cat was \\grining\\ from ear to ear.
Or"
                                        + " here to\n\n\n here.";
                System.out.println(speech);
                System.out.println(backSlashNewLine);


                //Looking for Who in The Who
                String bandName = "The Who";
                int index = bandName.indexOf("Who");

                System.out.println("I found Who at position " + index);

                String newBandName = bandName.substring(0,index);

                //The Who is now The Clash
                newBandName = newBandName + "Clash";
                System.out.println("Let's change the band name to " + newBandName);

                //Convert a string number to an actual number
                String number = "10";
                int convertedNumber = Integer.valueOf(number).intValue();
                System.out.println("The number " + convertedNumber);

                //Converting to a differnt number type.
                int numberTwenty = 20;
                String converted = Double.toString(numberTwenty);
                System.out.println(converted);

                //Time to trim some spaces
                String tooManySpaces = "     Neil Armstrong..   ";
                tooManySpaces = tooManySpaces.trim();


                //lexicographically Apple precedes Pear!
                String firstString = "Apple";
                String secondString = "Pear";

                if (firstString.compareTo(secondString) < 0){

                    System.out.println("Apple comes first!");

                }

            }
        }
```

```
Output:Here is Sherlock's address: I live at 22b Baker Street!
Char Array: Characters
Byte Array: Characters
```

```
Thomas Müller plays for Germany.
"I say old chap", he said to me
The cat was \grining\ from ear to ear. Or here to


 here.
I found Who at position 4
Let's change the band name to The Clash
The number 10
20.0
Apple comes first!
```

2.Example of String Concatination.




```
package com.tcs.ilp.stringdemo;

public class StringConcatExample {
      /*

    This Java String concat example shows how to concat String in Java.
*/

    public static void main(String args[]){
            /*
```

```
             * String concatenation can be done in several ways in Java.
             */

            String str1 = "Hello";
            String str2 = " World";

            //1. Using + operator
            String str3 = str1 + str2;
            System.out.println("String concat using + operator : " + str3);

            /*
             * Internally str1 + str 2 statement would be executed as,
             * new StringBuffer().append(str1).append(str2)
             *
             * String concatenation using + operator is not recommended for large
number
             * of concatenation as the performance is not good.
             */

            //2. Using String.concat() method
            String str4 = str1.concat(str2);
            System.out.println("String concat using String concat method : " +
str4);

            //3. Using StringBuffer.append method
            String str5 = new StringBuffer().append(str1).append(str2).toString();
            System.out.println("String concat using StringBuffer append method : "
+ str5);
    }

}
```

3.Example for comparing two Strings.

```
package com.tcs.ilp.stringdemo;

public class StringCompareExample {


          public static void main(String a[]){

              String c = "We are comparing the content with a StringBuffer
content";
              StringBuffer sb =
                  new StringBuffer("We are comparing the content with a
StringBuffer content");
              /**
               * We can use contentEquals() method to compare content with a
StringBuffer.
               * It returns boolean value.
               */
              if(c.contentEquals(sb)){
                  System.out.println("The content is equal");
              } else {
```

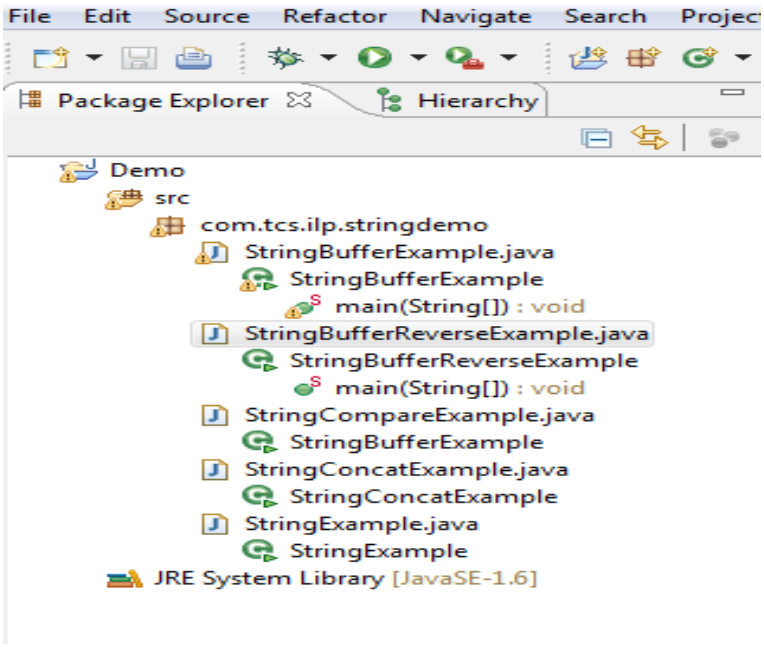
```
                System.out.println("The content is not equal");
            }
            StringBuffer asb =
                new StringBuffer("You cannot compare the content with a String
content");
            if(c.contentEquals(asb)){
                System.out.println("The content is equal");
            } else {
                System.out.println("The content is not equal");
            }
        }
    }
```

4.Example to create StringBuffer using Constructor.

```
package com.tcs.ilp.stringdemo;

public class StringBufferExample {

    /*
       Java StringBuffer Example
       This example shows how java StringBuffer can be built and created using
       different constructors of Java StringBuffer class.
    */
      public static void main(String[] args) {
        /*
          Java StringBuffer is a mutable sequence of characters.
          Difference between Java String and StringBuffer is that StringBuffer
          can be modified while String can not.
          Java StringBuffer can buit using one of the following constructors
        */

        /*
        1. StringBuffer StringBuffer()
        Construct empty StringBuffer with initial capacity of 16
        */

        StringBuffer sbObj1 = new StringBuffer();

        /*
        2. StringBuffer StringBuffer(int length)
        Constructs empty StringBuffer with initial capacity of length
        */
        StringBuffer sbObj2 = new StringBuffer(10);

        /*
        3. StringBuffer StringBuffer(String str)
        constructs StringBuffer with the contents same as argument String
        */
        StringBuffer sbObj3 = new StringBuffer("Hello World");
        System.out.println(sbObj3);

      }
```

```
}
```

5.Example to show reverse using StringBuffer.




```
package com.tcs.ilp.stringdemo;

public class StringBufferReverseExample {
	/*

	   This example shows how to reverse the content of the StringBuffer using
	   reverse method of Java StringBuffer class.
	*/


	  public static void main(String[] args) {

	    //create StringBuffer object
```

```
        StringBuffer sb = new StringBuffer("Java StringBuffer Reverse Example");
        System.out.println("Original StringBuffer Content : " + sb);

        //To reverse the content of the StringBuffer use reverse method
        sb.reverse();
        System.out.println("Reversed StringBuffer Content : " + sb);
      }
    }
```

6. Example for StringBuilder

```
package com.tcs.ilp.stringdemo;

//StringBuilder Example
public class StringBuilderExample {
    public static void main(String[] args) {

        // Create new StringBuilder.
        StringBuilder builder = new StringBuilder();

        // Loop and append values.
        for (int i = 0; i < 5; i++) {
            builder.append("abc ");
        }
        // Convert to string.
        String result = builder.toString();

        // Print result.
        System.out.println(result);
      }
    }
```

7. Given a string, compute a new string where identical chars that are adjacent in the original string are separated from each other by a "*".

8. Given a string, compute a new string where all the lowercase 'x' chars have been moved to the end of the string.

9.Write a Program to sort a String

10.Consider the following string:
String hannah = "Did Hannah see bees? Hannah did.";
What is the value displayed by the expression hannah.length()?
What is the value returned by the method call hannah.charAt(12)?

Write an expression that refers to the letter b in the string referred to by hannah.

11.Show two ways to concatenate the following two strings together to get the string "Hi, mom.":

```
String hi = "Hi, ";
String mom = "mom.";
```