Course code : **CSE1004**

Course title : **Problem Solving using Java**

Java – Strings

Objectives

This session will give the knowledge about to

- Work with String class
- How to use String predefined methods

Introduction to Strings

- String is a group of characters. They are objects of type String class.
- Once a String object is created it cannot be changed.
- Strings are Immutable.
- String are declared as final, so there cannot be subclasses of these classes.
- The default constructor creates an empty string.
    - String str = new String();

Introduction to Strings

To create a String in java is

```
String s1 = “welcome”;                    //primitive type
```

equivalent to

```
String s2 = new String(“welcome”);    //object type
```

or equivalent to

```
char data[] = {‘w’, ‘e’, ‘l’, ‘c’, ‘o’, ‘m’, ‘e’};        //array type
String s3 = new String(data);
```

or equivalent to

```
String s4 = new String(s1);               //reference type
```

String class - Methods

1. The length() method returns the length of the string.

   Ex: System.out.println("Varun".length());

   Output: prints 5

2. The + operator is used to concatenate two or more strings.

   Ex: String myName = "Varun";

   String s = "My name is" + myName+ ".";

   For string concatenation the Java compiler converts an operand to a String whenever the other operand of the + is a String object.

String class - Methods

3. charAt() method. Characters in a string can be retrieved in a number of ways

public char **charAt**(int index)

- Method returns the character at the specified index.
- An index ranges from 0 to length()-1

```
char   c;
c = "abc".charAt(1);    // c = ‘b’
```

String class - Methods

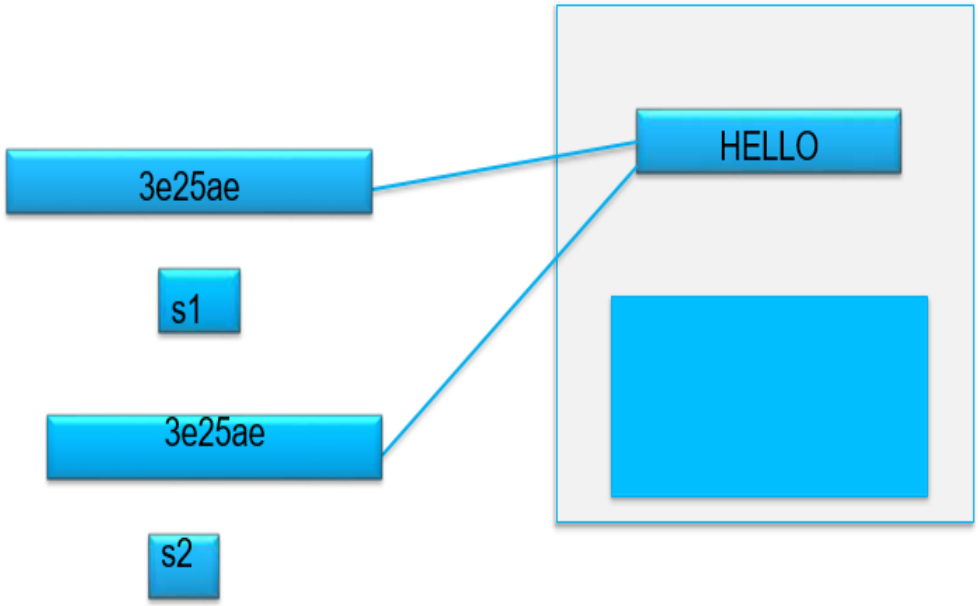
4. equals() Method- This method is used to compare Strings. It will return true, if the argument is not null and it contains the same sequence of characters.
   - **public boolean equals(anotherString);**
   - String s1="Welcome"; String s2="welcome";
   - boolean result = s1.equals(s2); //false is the result

5. equalsIgnoreCase() Method- Compares this String to another String, ignoring case considerations..
   - **public boolean equalsIgnoreCase(anotherString);**
   - String s1="Welcome"; String s2="welcome";
   - boolean result = s1.equalsIgnoreCase(s2); //true is the result

String Comparison

String s1 = “HELLO”;

String s2 = “HELLO”;




```
String s1=“HELLO”;
String s2=“HELLO”;
if(s1==s2)
        System.out.println(“equal”);
else
        System.out.println(“ not equal”);
```

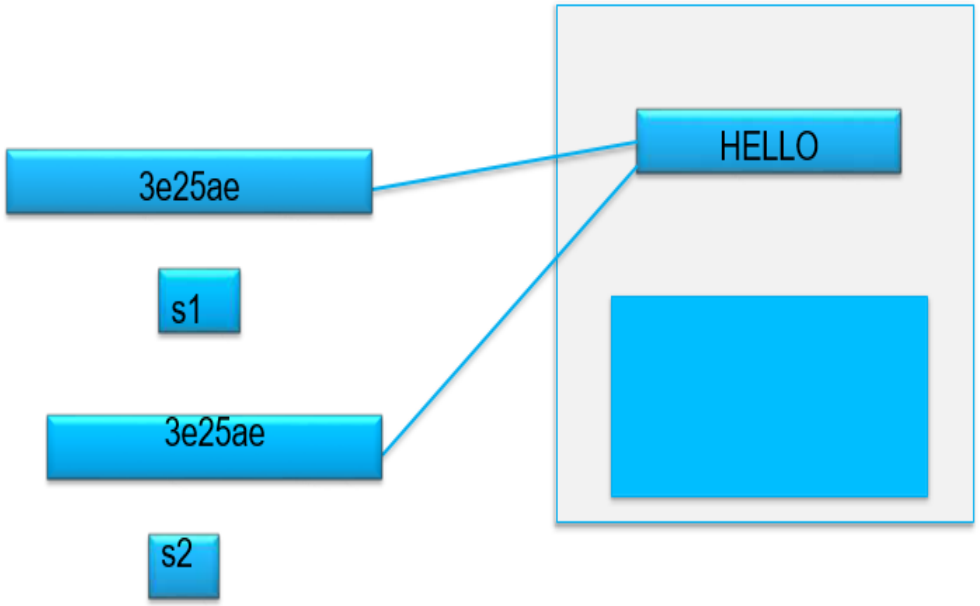
Output:

equal

String Comparison

String s1 = “HELLO”;

String s2 = “HELLO”;




```
String s1=“HELLO”;
String s2=“HELLO”;
if(s1.equals(s2))
        System.out.println(“equal”);
else
        System.out.println(“ not equal”);
```

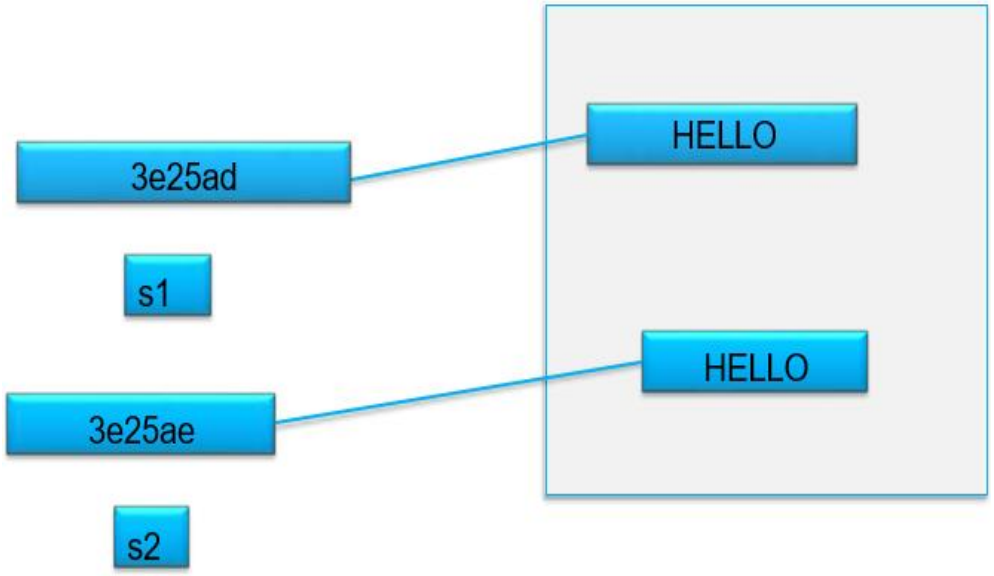
Output:

equal

String Comparison

String s1 = new String(“HELLO”);

String s2 = new String(“HELLO”);




```
String s1=“HELLO”;
String s2=“HELLO”;
if(s1 == s2)
        System.out.println(“equal”);
else
        System.out.println(“ not equal”);
```

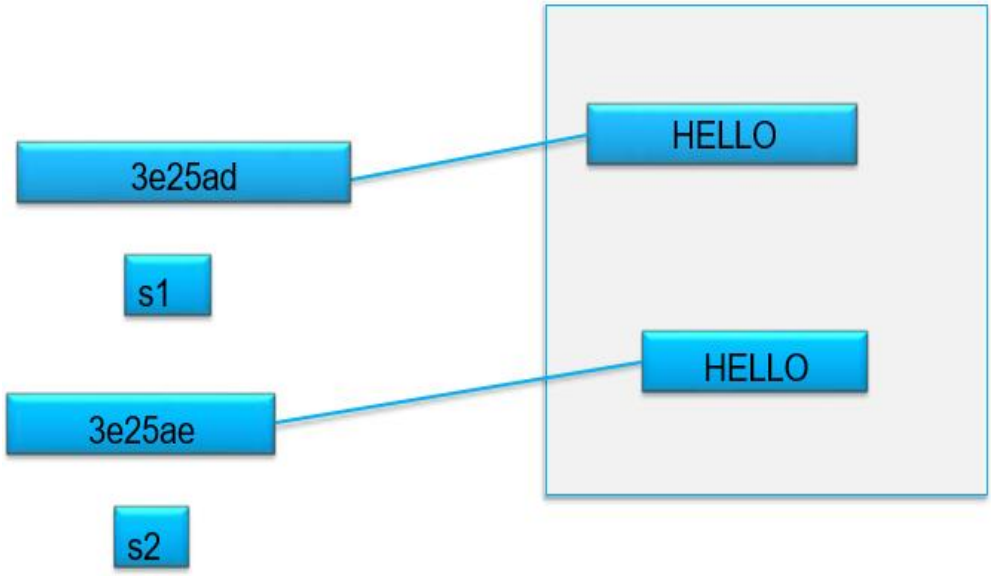
Output:

not equal

String Comparison

String s1 = new String(“HELLO”);

String s2 = new String(“HELLO”);




```
String s1=“HELLO”;
String s2=“HELLO”;
if(s1.equals(s2))
        System.out.println(“equal”);
else
        System.out.println(“ not equal”);
```

Output:

equal

String class Methods

6. startsWith() – Tests if this String starts with the specified prefix.

   public boolean startsWith(String prefix)

   "January".startsWith("Jan");

   // Output: true

7. endsWith() - Tests if this String ends with the specified suffix.

   public boolean endsWith(String suffix)

   "January".endsWith("ry");

   // Output: true

String class Methods

8. **compareTo()** - Compares two strings and to know which string is bigger or smaller
   - We will get a negative integer, if this String object is less than the argument string
   - We will get a positive integer if this String object is greater than the argument string.
   - We will get a return value 0(zero), if these strings are equal.

   public int **compareTo**(String anotherString)

9. public int **compareToIgnoreCase**(String str)

   This method is similar to compareTo() method but this does not take the case of strings into consideration.

String class Methods

10. indexOf() Method

Searches for the first occurrence of a character or substring. Returns -1 if the character does not occur

- **public int indexOf(int ch)** - It searches for the character represented by ch within this string and returns the index of first occurrence of this character
- **public int indexOf(String str)** - It searches for the substring specified by str within this string and returns the index of first occurrence of this substring

Example:

```
String str = “How was your day today?”;
str.indexOf(‘t’);      //17
str.indexOf(“was”);  //4
```

String class Methods

- **public int indexOf(int ch, int index)** - It searches for the character represented by ch within this String and returns the index of first occurrence of this character starting from the position specified by from index

- **public int indexOf(String str, int index)** - It searches for the substring represented by str within this String and returns the index of first occurrence of this substring starting from the position specified by from index

  Example:

```
String str = “How was your day today?”;
str.indexOf(‘a’,6);      //14
str.indexOf(“was”,2);  //4
```

String class Methods

11. lastIndexOf() – It searches for the last occurrence of a particular character or substring
12. substring() - This method returns a new string which is actually a substring of this string. It extracts characters starting from the specified index all the way till the end of the string

public String substring(int beginIndex)

Eg: “unhappy”.substring(2) returns “happy”

public String substring(int beginIndex, int endIndex)

Eg: “smiles”.substring(1,5) returns “mile”

String class Methods

13. concat() - Concatenates the specified string to the end of this string

public String concat(String str)

"to".concat("get").concat("her") //return together

14. replace() - Returns a new string resulting from replacing all occurrences of oldChar in this string with newChar

public String replace(char oldChar, char newChar);

```
String str = "How was your day today?";
System.out.println(str.replace('a', '3'));     //displays How w3s your d3y tod3y?
```

String class Methods

15. trim() **-** Returns a copy of the string, with leading and trailing whitespace omitted

public String trim()

String s = “Hi Mom! “.trim();

S = “Hi Mom!”

16. valueOf() **–** This method is used to convert a character array into String. The result is a String representation of argument passed as character array

public static String valueOf (char[] data);

String class Methods

valueOf() – This method is used to convert anything into String.

- String s= String.valueOf(boolean b);
- String s= String.valueOf(char c);
- String s= String.valueOf(int i);
- String s= String.valueOf(float f);
- String s= String.valueOf(double d);

String class Methods

17. toLowerCase(): Method converts all of the characters in a String to lower case

    public String toLowerCase();

    String s = “JaVa”.toLowerCase(); // java

18. toUpperCase(): Method converts all of the characters in a String to upper case

    public String toUpperCase();

    String s = “JaVa”.toLowerCase(); // JAVA

Quiz

Which of these class is superclass of String class?

a) java.util

b) java.lang

c) ArrayList

d) None of the mentioned

Quiz

Which of these operators can be used to concatenate two or more String objects?

a) +

b) +=

c) &

d) ||

Quiz

Which of this method of class String is used to obtain a length of String object?

a) get()

b) Sizeof()

c) lengthof()

d) length()

Quiz

Which of these method of class String is used to extract a single character from a String object?

a) CHARAT()

b) chatat()

c) charAt()

d) ChatAt()

Quiz

Which of these constructors is used to create an empty String object?

a) String()

b) String(void)

c) String(0)

d) None of the mentioned

Quiz

What will be the output of the following Java program?

```
class String_demo
  {
     public static void main(String args[])
     {
        char chars[] = {'a', 'b', 'c'};
        String s = new String(chars);
        System.out.println(s);
     }
  }
```

Quiz

What will be the output of the following Java program?

```
class String_demo   {
        public static void main(String args[])  {
            char chars[] = {'a', 'b', 'c'};
            String s = new String(chars);
            String s1 = "abcd";
            int len1 = s1.length();
            int len2 = s.length();
            System.out.println(len1 + " " + len2);
        }
  }
```

Quiz

Online link for MCQ Practice

https://www.examveda.com/java-program/practice-mcq-question-on-strings/

Example

```
public class Main {
    public static void main(String arg[]) {
     String s="Welcome";
     System.out.println(s);
     System.out.println("1."+s.length());
     System.out.println("2."+s+" + example");
     System.out.println("3."+s.charAt(5));
     System.out.println("4."+s.equals("Welcome"));
     System.out.println("5."+s.equalsIgnoreCase("welcome"));
     System.out.println("6."+s.startsWith("Wel"));
```

Example

```
System.out.println("7."+s.endsWith("ava"));
System.out.println("8."+s.compareTo("welcome"));
System.out.println("9."+s.compareToIgnoreCase("welcome"));
System.out.println("10."+s.indexOf('e'));
System.out.println("10."+s.indexOf("me"));
System.out.println("10."+s.indexOf('e',3));
System.out.println("11."+s.lastIndexOf('e'));
System.out.println("12."+s.substring(3));
System.out.println("12."+s.substring(3,5));
```

Example

```
System.out.println("13."+s.concat("    "));
s=s.concat("    ");
System.out.println("14."+s.replace('e','s'));
System.out.println("15."+s.length());
s = s.trim();
System.out.println("15."+s.trim());
System.out.println("15."+s.length());
char[] data={'J', 'a', 'V', 'a'};
s = String.valueOf(data);
```

Example

```
        System.out.println("16."+s);
        System.out.println("17."+s.toLowerCase());
        System.out.println("18."+s.toUpperCase());
      }
}
```

Output

Welcome

1.7

2.Welcome + example

3.m

Example

4.true

5.true

6.true

7.false

8.-32

9.0

10.1

10.5

10.6

11.6

Example

12.come

12.co

13.Welcome

14.Wslcoms

15.11

15.Welcome

15.7

16.JaVa

17.java

18.JAVA

Summary

We have discussed about

- Work with String class
- How to use String predefined methods