

String In Java (Part II)

- The `getBytes` method

`Java.getBytes()` method in `java` is used to convert a string into sequence of bytes and returns an array of bytes.

Example:

```
public class JavaApplication52 {  
    public static void main(String[] args) {  
        String s="ABCDEF";  
        byte[] b= s.getBytes();  
        for (int i=0;i<b.length;i++)  
            System.out.println(b[i]);  
    }  
}
```

The output is:

65

66

67

68

69

70

If we replace the string to `s="abcdef"`;

The output will be:

97

98

99

100

101

102

H.W. what is the output when the string s="0123456"

LEC10 OOP 2018-2019

- **The toCharArray() method**

The **java** string **toCharArray()** method converts the given string into a sequence of characters. The returned array length is equal to the length of the string.

Example:

```
public class JavaApplication53 {  
    public static void main(String[] args) {  
        String s="abcdef";  
        char[] c= s.toCharArray();  
        for (int i=0;i<c.length;i++)  
            System.out.println(c[i]);  
    }  
}
```

The output is:

a
b
c
d
e
f

if we replace the string s to “123456”

The output will be :

1
2
3
4
5
6

- **The equals() method**

The **java** string **equals()** method compares the two given strings based on the content of the string. If any character is not matched, it returns false. If all characters are matched, it returns true.

LEC10 OOP 2018-2019

Example:

```
public class EqualsExample {
    public static void main(String[] args) {
        String s1 = "javatpoint"; String s2 = "javatpoint";
        String s3 = "Javatpoint";
        System.out.println(s1.equals(s2)); // True because content is same
        if (s1.equals(s3)) { System.out.println("both strings are equal"); }
        else System.out.println("both strings are unequal");    } }
```

The output is:

true

both strings are unequal

- The equalsIgnoreCase() method

The **String equalsIgnoreCase()** method compares the two given strings on the basis of content of the string irrespective of case of the string. It is like equals() method but doesn't check case. If any character is not matched, it returns false otherwise it returns true.

Example:

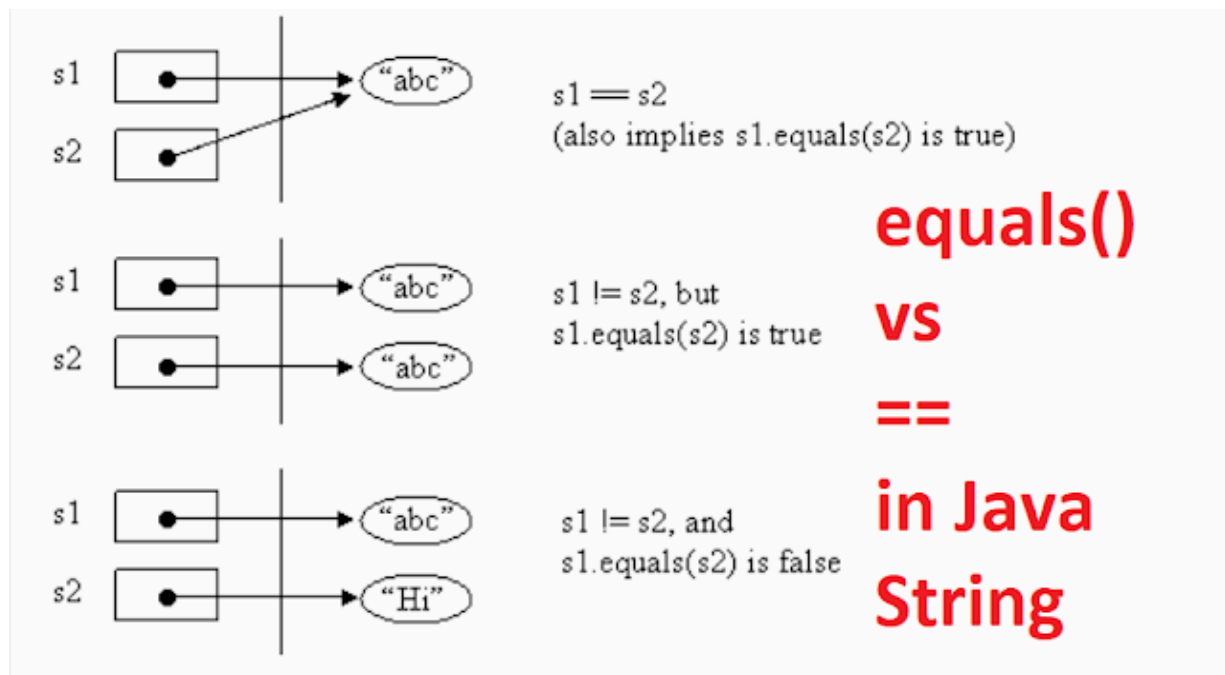
```
public class EqualsIgnoreCaseExample{
    public static void main(String args[]){
        String s1="javatpoint"; String s2="javatpoint";
        String s3="JAVATPOINT"; String s4="python";
        System.out.println(s1.equalsIgnoreCase(s2));
        System.out.println(s1.equalsIgnoreCase(s3));
        System.out.println(s1.equalsIgnoreCase(s4));}}
```

The output is:

true

true

false



- The `startsWith()` method

This method checks if this string starts with given prefix. It returns true if this string starts with given prefix else returns false. It has two forms:

- 1- **boolean `startsWith(String str)`**: It returns true if the `str` is a prefix of the String.
- 2- **boolean `startsWith(String str, index fromIndex)`**: It returns true if the String begins with `str`, it starts looking from the specified index “`fromIndex`”.

Example:

```
public class Main{
    public static void main(String[] args) {
        String s="Yasser Mohammed Ali";
        System.out.println(s.startsWith("Ya"));
        System.out.println(s.startsWith("ya"));
        System.out.println(s.startsWith("Yasser"));
        System.out.println(s.startsWith("Y"));
        System.out.println(s.startsWith("Ali"));
        System.out.println(s.startsWith("Ali",16));}}

```

The output is:

```
true
false
true
true
false
true
```

LEC10 OOP 2018-2019

- The `endsWith()` method

This method checks if this string ends with given suffix. It returns true if this string ends with given suffix else returns false. The form is:

boolean endsWith(String str): It returns true if the str is a suffix of the String.

Example:

```
public class JavaApplication52 {
    public static void main(String[] args) {
        String s="Yasser Mohammed Ali";
        System.out.print(s.endsWith("Ali"));
        System.out.print(s.endsWith("Yasser"));
        System.out.print(s.endsWith("i"));
        System.out.print(s.endsWith("li"));
        System.out.print(s.endsWith("er"));  }}
```

The output is:

Truefalsetruefalse

- The Java String `replace()`

This method returns a string replacing all the old char or CharSequence to new char or CharSequence..

There are two type of replace methods in java string.

- 1- `public String replace(char oldChar, char newChar).`
- 2- `public String replace(CharSequence target, CharSequence replacement)`

Example:

```
public class ReplaceExample1{
    public static void main(String args[]){
        String s1="aaabbbccc";
        String s2=s1.replace('a','e');
        System.out.println(s1+" "+s2);
        String s3=s1.replace("ab","zx");
        System.out.println(s1+" "+s3); }}
```

The output is:

aaabbbccc eeabbbccc

aaabbbccc aazxbbccc

LEC10 OOP 2018-2019

- **The CompareTo() method**

It is used for comparing two strings lexicographically. Each character of both strings is converted into a Unicode value. Assume that you have two strings a1 and a2, this method returns:

if a1 > a2, it returns negative number

if a1 < a2, it returns positive number

if a1 == a2, it returns 0

Example:

```
public class JavaApplication52 {
    public static void main(String[] args) {
        String s1="AA"; String s2="AAA ";
        String s3="BBB", s4="BB";
        String s5="aaa";
        System.out.println(s1.compareTo(s2));   System.out.println(s1.compareTo(s3));
        System.out.println(s4.compareTo(s2));   System.out.println(s1.compareTo(s2));
        System.out.println(s1.compareTo(s4)); System.out.println(s5.compareTo(s2)); }}
```

The output is:

```
-1
-1
1
-1
-1
32
```

H.W. Trace the following program:

```
public class CompareToExample {
    public static void main(String args[]) {
        String str1 = "String method tutorial";String str2 = "compareTo method example";
        String str3 = "String method tutorial";
        int var1 = str1.compareTo( str2 );
        System.out.println("str1 & str2 comparison: "+var1);
        int var2 = str1.compareTo( str3 );
        System.out.println("str1 & str3 comparison: "+var2);
        int var3 = str2.compareTo("compareTo method example");
        System.out.println("str2 & string argument comparison: "+var3);  }}
```

Output:

```
str1 & str2 comparison: -16
```

```
str1 & str3 comparison: 0
```

```
str2 & string argument comparison: 0
```

The CompareToIgnoreCase() method: This method compares two strings lexicographically, ignoring case differences. This means that “aa” equals “AA”.

Converting from any data type to String

- 1- It is very simple way to convert any data type to a string using the + operator as shown at the following java code:

Example:

```
public class JavaApplication52 {  
    public static void main(String[] args) {  
        int i=1975;  
        String s1=i+"";  
        System.out.println(i);  
        System.out.println(s1);  
    }  
}
```

The output is:

```
1975  
1975
```

Example

```
public class JavaApplication52 {  
    public static void main(String[] args) {  
        int i=1975;  
        String s1=i+"";  
        System.out.println(i);  
        System.out.println(s1);  
        i++; System.out.println(i);  
        s1++; System.out.println(s1);  
    }  
}
```

The output is:

```
????????????? why?
```

2- The ValueOf() method

This method converts different types of values into string. Using this method, you can convert int to string, long to string, Boolean to string, character to string, float to string, double to string, object to string and char array to string. This method is static method, so we can call it using String class directly.

```
valueOf(boolean) ,valueOf(char c),valueOf(char[] c),valueOf(int i),valueOf(long l),  
valueOf(float f), valueOf(double d)
```

```
String s1; int i=1976;  
s1=String.valueOf(i); System.out.println(i); System.out.println(s1);
```

3- The toString method

It could be converted any data type to string using toString method, for example to convert integer (the same to double, long, float ...etc.) to string is shown below:

```
int i = 42;  
String str = Integer.toString(i);
```

LEC10 OOP 2018-2019

`System.out.println(i); System.out.println(str);`

Converting from String to any Data Type

It could be converted any string to its corresponding data type using `valueOf` method or `parseInt`, for example to convert a string that holds integer value to integer variable is shown below:

1- String to integer using `valueOf()` method

```
String str = "25";  
int i = Integer.valueOf(str).intValue();  
i++; System.out.println(i);
```

2- String to integer using `parseInt()` method

```
String str = "25";  
int i = Integer.parseInt(str);  
i++; System.out.println(i);
```

the same method could be used to convert from string to double, float, long ...etc.

what is the output and Why?

```
String str = "25.8";  
int i = Integer.parseInt(str);  
i++; System.out.println(i);
```


Problems

- 1- Define a list of names and sorting them alphabetically.
- 2- Find a number of digits before and after decimal point at any double number.
- 3- Converting any upper case letter to lower case letter and verse versa in a string "IrAQ"
- 4- Print a list of names that starts with an upper case letter only.
- 5- Print a list of names that starts with 'A' or 'a' letter only.
- 6- Reversing (عكس) any string , example string s="abcd", the output is string "dcba"
- 7- Consider the following string:

String hannah = "Did Hannah see bees? Hannah did.";

- a. What is the value displayed by the expression hannah.length()?
 - b. What is the value returned by the method call hannah.charAt(12)?
 - c. Write an expression that refers to the letter b in the string referred to by hannah.
- 8- How long is the string returned by the following expression? What is the string?
- "Was it a car or a cat I saw?".substring(9, 12)