Strings

A string is a sequence of characters

A string is initialized as follows:

```
String name;
name = new String("Harry");
```

String is a data type & can~~not~~ be ~~changed~~ used like a data type
Strings are immutable & cannot be changed:

String name = "Harry";
reference → object

Different ways to print in Java
We can use the followings ways to print in Java:

1) System.out.print() → No newline at end
2) System.out.println() → Newline at the end
3) System.out.printf() } work in the same
4) System.out.format() } way

For e.g. System.out.printf("%c", ch);
%d for integers
%f for float
%c for char
%s for string

String methods

```
        01234
String name = "Harry";
```

1) name.length() → Returns length of string name (5 in this case)

2) name.toLowerCase() → Returns a new string which has all the lowercase characters from the string name & converts high to low case.

3) name.toUpperCase() → Returns a new string which has all the upper case characters from the string name & converts other characters to upper case

4) name.trim() → Returns a new string after removing all the leading & trailing spaces from the original string.

5) name.substring(int start) → Returns a substring from start to the end. substring(3) returns "ry". (Note: Index starts from zero)

6) name.substring(int start, int end) Returns a substring from start index to the end index. Start index is included & end is excluded.

7) name.replace('r', 'p') →
Returns a new string after replacing r with p. Happy is returned in case.

8) name.startsWith("Ha") →
Returns true if name starts with string "Ha". true in this case.

9) name.endsWith("ry") →
Returns true if name ends with string "ry". true in this case.

10) name.charAt(2) →
Returns character at a given index position, r in this case.

11) name.indexOf(str) →
Returns index of the given string for e.g. name.indexOf("ar") returns 1 which is the first occurence of ar in string "Harry", -1 otherwise.

12) name.indexOf("s", 3) →
Returns index of given string starting from the index 3 (int), -1 returned in this case.

13) name.lastIndexOf("r") →
Returns the last index of given string, 3 in this case.

14) name.lastIndexOf("r", 2) →
Returns the last index of given string before index 2.

15) name.equals("Harry") →
Returns true if the given string is equal to "Harry", false otherwise (case sensitive)

16) name.equalsIgnoreCase("harry") →
Returns true if strings are equal ignoring case of characters.

Escape Sequence Characters

Sequence of characters after backslash \ is Escape sequence characters.

Escape sequence characters consist of more than one characters but represents one character when used within strings

\t → New tab

\b → Insert backspace

\n → Newline

\r → Insert carriage return
\f → Insert form feed
\' / \" → Insert single/double quote
\\ → Insert backslash

'+' is used to concatenate 2 strings in Java

Practice set

Q1 Java program to convert a string to lower case

```
package com.company

public class String{

    psvm(String[] args)
    {
        String name= "Adwaitp";
        sout(
        name= name.toLowerCase();
        sout(name);
    }
}
```

O/p

adwaitp

Q2 Program to replace spaces with underscores.

```
public class String{

    psvm(String args[])
    { String text= "To Lower Case";
      text = text.replace(" ", "_");
      sout(text);
    }
}
```

To_Lower_Case