COMSATS UNIVERSITY, ISLAMABAD



**Programming Fundamentals**

--CSC103--

**Name**: Talha Rizwan

**Reg no**. SP20-BSE-093

**Section**: B

**Submitted to:** Dr. Behjat Zuhaira

*Date*: 10/08/2020

# System.out.printf():

In Java printf() method is used for formatting of string just like in other programming languages.

## Syntax:

System.out.printf(format, arguments);

# String formatting:

To format a string, we use %s combination in the format section of printf() syntax.

Following are some formatting styles for string.

## Uppercase string:

We can make a lowercase string to uppercase string using following Syntax.

System.out.printf(“ %S ”,” arguments”);

### Example:

public class Format {

public static void main(String[] args) {

System.out.printf("%S ","hello");

}

}

Output:

“HELLO”

Graphical user interface, application, Word

Description automatically generated

## Specifying Minimum length:

### Right Justify:

We can specify a specific length for a string.

Syntax:

System.out.printf(“ % ‘length’s ” , “ arguments “);

### Example:

public class Format {

public static void main(String[] args) {

System.out.printf(" %20s ","Hello World");

}

}

Output:

“ Hello World”

Graphical user interface, application, Word

Description automatically generated

It will right justify the string and leave the blank spaces at left.

### Left Justify:

If you want to left justify the string use the following syntax.

System.out.printf(“ % -‘length’s ” , “ arguments “);

### Example:

public class Format {

public static void main(String[] args) {

System.out.printf(" %-20s ","Hello World");

}

}

Output:

“Hello World ”

Graphical user interface, application, Word

Description automatically generated

## Limit Number of Words:

To precise the number of words we can format it with the syntax.

System.out.printf(“ % X.Ys ” , “ arguments “);

X is padding and Y is the number of characters

### Example:

public class Format {

public static void main(String[] args) {

System.out.printf("%1.5s ","Hello World");

}

}

Output:

“Hello”

Graphical user interface, application, Word

Description automatically generated

# Boolean Formatting:

In Boolean Formatting we use %b format. If the output is true it will show true otherwise it will show false

Syntax:

System.out.printf(“ %b”,”argument”);

Example:

public class Format {

public static void main(String[] args) {

System.out.printf("%b ","Hello");

}

}

Output:

“True”

Graphical user interface, application, Word

Description automatically generated

We can also uppercase the alphabets of true by using capital B.

Example:

public class Format {

public static void main(String[] args) {

System.out.printf("%B ",null);

}

}

Output:

“FALSE”

Graphical user interface, application, Word

Description automatically generated

# Date and Time Formatting:

In java we can format date and time using printf(). As before we can use different characters in printf() syntax.

## Time Formatting:

To show the time on console we can use %tT with the date method to show the current time.

There are some specail characters for formatting.

* ‘H', ‘M', ‘S'  – characters are responsible for extracting the hours, minutes and second from the input Date
* ‘L', ‘N'  – to represent the time in milliseconds and nanoseconds accordingly
* ‘p' – adds am/pm formatting
* ‘z' – prints out the timezone offset

Syntax:

System.out.printf(“%tT”,date);

Example:

import java.util.Date;

public class Format {

public static void main(String[] args) {

Date date = new Date();

System.out.printf(" %tT",date);

}

}

Output:

“10:20:36”

Graphical user interface, application, Word

Description automatically generated

## Detail Time Format:

For detailed time formatting we use the following syntax

System.out.printf(“hours %tH, minutes %tM , seconds %tS)”,date,date,date);

Example:

import java.util.Date;

public class Format {

public static void main(String[] args) {

Date date = new Date();

System.out.printf("hours %tH, minutes %tM, Seconds %tS",date, date ,date);

}

}

Output:

“hours 10, minutes 29, Seconds 45”

Graphical user interface, text, application

Description automatically generated

## $ function in Time Formatting:

As we saw in previous example that we put 3 saperate arguments for Hours, Minutes, and Seconds but using $ specifier we can specife a single argument to whole format.

Example:

import java.util.Date;

public class Format {

public static void main(String[] args) {

Date date = new Date();

System.out.printf("%1$tH:%1$tM:%1$tS %1$tp", date);

}

}

Output:

10:36:46 am

Graphical user interface, application, Word

Description automatically generated

### Example of other characters:

import java.util.Date;

public class Format {

public static void main(String[] args) {

Date date = new Date();

System.out.printf(" %1$tL %1$tN %n",date);

System.out.printf(" %tp %n",date);

System.out.printf(" %tz",date);

}

}

Output:

“656 656000000

am

+0500”

Graphical user interface, text, application

Description automatically generated

## Date Formatting:

As in Time formatting, we use special different characters for formatting. In Date formatting there are also some special characters for date formatting

* ‘A' – prints out the full day of the week
* ‘d' – formats a two-digit day of the month
* ‘B' – is for the full month name
* ‘m' – formats a two-digit month
* ‘Y' – outputs a year in four digits
* ‘y' – outputs the last two digits of the year

### Example:

import java.util.Date;

public class Format {

public static void main(String[] args) {

Date date = new Date();

System.out.printf("%tA %n", date);

System.out.printf("%td %n", date);

System.out.printf("%tB %n", date);

System.out.printf("%tm %n", date);

System.out.printf("%tY %n", date);

System.out.printf("%ty %n", date);

}

}

Output:

Thursday

08

October

10

2020

20

Graphical user interface, application

Description automatically generated