Java printf( ) Method Quick Reference

https://www.cs.colostate.edu/~cs160/.Summer16/resources/Java_printf_method_quick_reference.pdf

```
System.out.printf( “format-string” [, arg1, arg2, … ] );
```

Format String:

Composed of literals and format specifiers. Arguments are required only if there are format specifiers in the format string. Format specifiers include: flags, width, precision, and conversion characters in the following sequence:

% [flags] [width] [.precision] conversion-character (square brackets denote optional parameters)

Flags:

- : left-justify (default is to right-justify)

+ : output a plus (+) or minus (-) sign for a numerical value

0 : forces numerical values to be zero-padded (default is blank padding)

, : comma grouping separator (for numbers > 1000)

: space will display a minus sign if the number is negative or a space if it is positive

Width:

Specifies the field width for outputting the argument and represents the minimum number of characters to be written to the output. Include space for expected commas and a decimal point in the determination of the width for numerical values.

Precision:

Used to restrict the output depending on the conversion. It specifies the number of digits of precision when outputting floating-point values or the length of a substring to extract from a String. Numbers are rounded to the specified precision.

Conversion-Characters:

d : decimal integer (byte, short, int, long)

f : floating-point number (float, double)

c : character (Capital C will uppercase the letter)

s : String (Capital S will uppercase all the letters in the string)

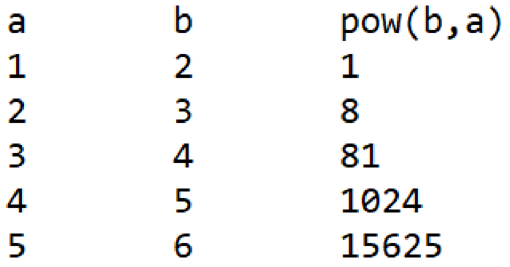
h : hashcode (A hashcode is like an address. This is useful for printing a reference)

n : newline (Platform specific newline character- use %n instead of \n for greater compatibility)

Example

```
System.out.printf("%-8s%-8s%-8s\n","a","b","pow(b,a)");

System.out.printf("%-8d%-8d%-8d\n",1,2,1);

System.out.printf("%-8d%-8d%-8d\n",2,3,8);

System.out.printf("%-8d%-8d%-8d\n",3,4,81);

System.out.printf("%-8d%-8d%-8d\n",4,5,1024);

System.out.printf("%-8d%-8d%-8d\n",5,6,15625);
```

The above code will output a very nice table:



```
a       b       pow(b,a)
1       2       1
2       3       8
3       4       81
4       5       1024
5       6       15625
```