

# Formatted output in Java

Difficulty Level : Medium Last Updated : 27 Aug, 2021

Sometimes in Competitive programming, it is essential to print the output in a given specified format. Most users are familiar with printf function in C. Let us discuss how we can format the output in Java:

## Formatting output using System.out.printf()

This is the easiest of all methods as this is similar to printf in C. Note that System.out.print() and System.out.println() take a single argument, but printf() may take multiple arguments.

---

```
// A Java program to demonstrate working of printf() in Java
class JavaFormatter1
{
    public static void main(String args[])
    {
        int x = 100;
        System.out.printf("Printing simple integer: x = %d\n", x);

        // this will print it upto 2 decimal places
        System.out.printf("Formatted with precision: PI = %.2f\n", Math.PI);

        float n = 5.2f;

        // automatically appends zero to the rightmost part of decimal
        System.out.printf("Formatted to specific width: n = %.4f\n", n);

        n = 2324435.3f;

        // here number is formatted from right margin and occupies a
        // width of 20 characters
        System.out.printf("Formatted to right margin: n = %20.4f\n", n);
    }
}
```

## Output

```
Printing simple integer: x = 100
Formatted with precision: PI = 3.14
Formatted to specific width: n = 5.2000
Formatted to right margin: n =          2324435.2500
```

System.out.format() is equivalent to printf() and can also be used.

## Formatting using DecimalFormat class:

DecimalFormat is used to format decimal numbers.

---

```
// Java program to demonstrate working of DecimalFormat
import java.text.DecimalFormat;

class JavaFormatter2
{
    public static void main(String args[])
    {
        double num = 123.4567;

        // prints only numeric part of a floating number
        DecimalFormat ft = new DecimalFormat("####");
        System.out.println("Without fraction part: num = " + ft.format(num));

        // this will print it upto 2 decimal places
        ft = new DecimalFormat("#.##");
        System.out.println("Formatted to Give precision: num = " + ft.format(num));

        // automatically appends zero to the rightmost part of decimal
        // instead of #,we use digit 0
        ft = new DecimalFormat("#.000000");
        System.out.println("appended zeroes to right: num = " + ft.format(num));

        // automatically appends zero to the leftmost of decimal number
        // instead of #,we use digit 0
        ft = new DecimalFormat("00000.00");
        System.out.println("formatting Numeric part : num = "+ft.format(num));

        // formatting money in dollars
        double income = 23456.789;
        ft = new DecimalFormat("$###,###.##");
        System.out.println("your Formatted Dream Income : " + ft.format(income));
    }
}
```

## Output

```
Without fraction part: num = 123
Formatted to Give precision: num = 123.46
appended zeroes to right: num = 123.456700
formatting Numeric part : num = 00123.46
your Formatted Dream Income : $23,456.79
```

## Formatting dates and parsing using SimpleDateFormat class:

This class is present in java.text package.

---

```
// Java program to demonstrate working of SimpleDateFormat
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;

class Formatter3
{
    public static void main(String args[]) throws ParseException
    {
        // Formatting as per given pattern in the argument
        SimpleDateFormat ft = new SimpleDateFormat("dd-MM-yyyy");
        String str = ft.format(new Date());
        System.out.println("Formatted Date : " + str);

        // parsing a given String
        str = "02/18/1995";
        ft = new SimpleDateFormat("MM/dd/yyyy");
        Date date = ft.parse(str);

        // this will print the date as per parsed string
        System.out.println("Parsed Date : " + date);
    }
}
```

## Output:

Formatted Date : 09-08-2018

Parsed Date : Sat Feb 18 00:00:00 UTC 1995

## References:

<https://docs.oracle.com/javase/tutorial/essential/io/formatting.html>

<https://docs.oracle.com/javase/tutorial/java/data/numberformat.html>

<http://docs.oracle.com/javase/6/docs/api/java/text/SimpleDateFormat.html>

This article is contributed by **Pankaj Kumar**. If you like GeeksforGeeks and would like to contribute, you can also write an article using [write.geeksforgeeks.org](https://write.geeksforgeeks.org) or mail your article to [review-team@geeksforgeeks.org](mailto:review-team@geeksforgeeks.org). See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.