Java Formatting Output Tutorial

Contents

Time required: 45 minutes

Online Tutorial

Go through the following tutorials.

- Java String Format

Java String Format Specifiers

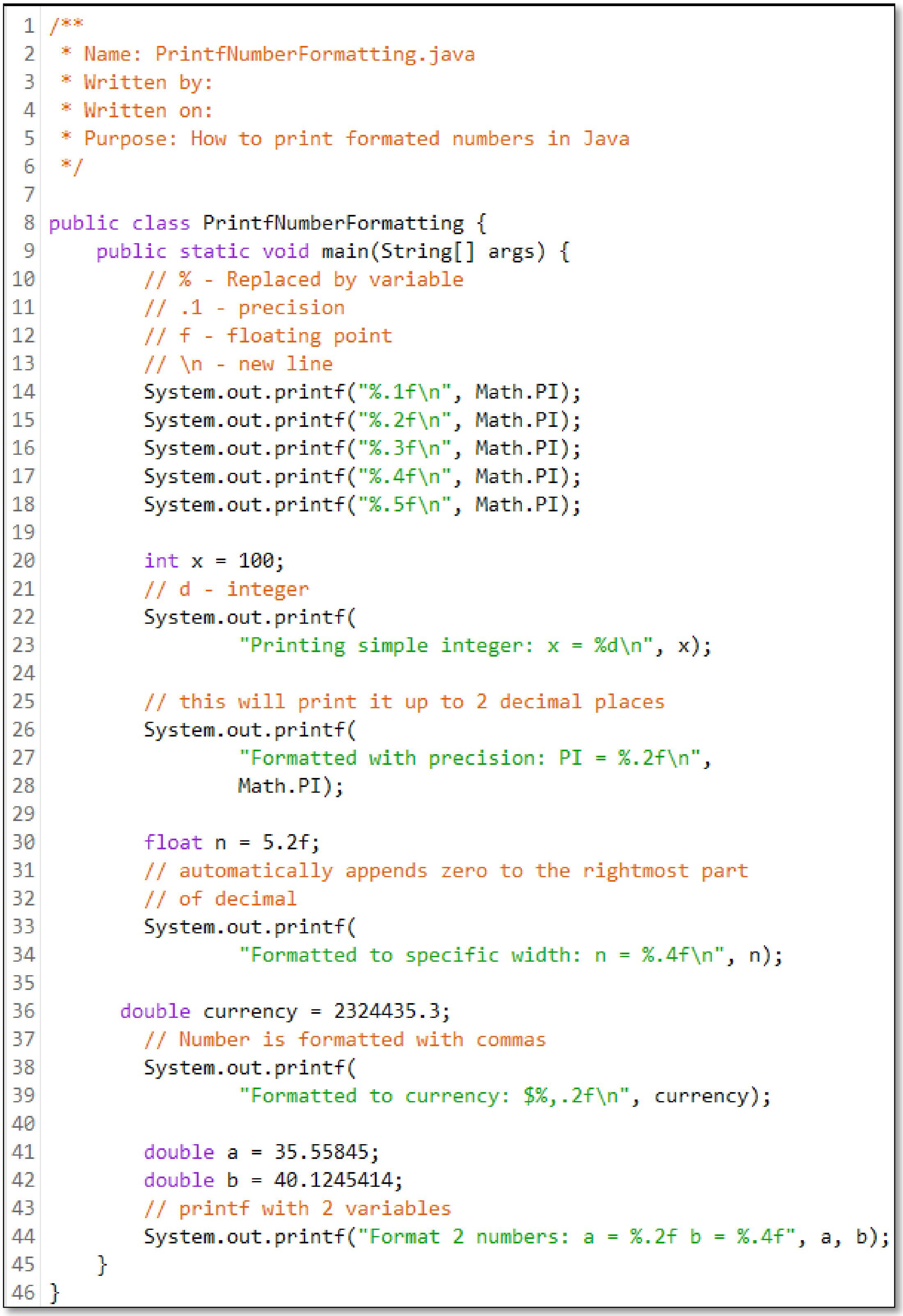
Format Specifier	Data Type	Output
%a	floating point (except BigDecimal)	Returns Hex output of floating point number.
%b	Any type	"true" if non-null, "false" if null
%c	character	Unicode character
%d	integer (incl. byte, short, int, long, bigint)	Decimal Integer
%e	floating point	decimal number in scientific notation
%f	floating point	decimal number

%g	floating point	decimal number, possibly in scientific notation depending on the precision and value.
%h	any type	Hex String of value from hashCode() method.
%n	none	Platform-specific line separator.
%o	integer (incl. byte, short, int, long, bigint)	Octal number
%s	any type	String value
%t	Date/Time (incl. long, Calendar, Date and TemporalAccessor)	%t is the prefix for Date/Time conversions. More formatting flags are needed after this. See Date/Time conversion below.
%x	integer (incl. byte, short, int, long, bigint)	Hex string.

Tutorial 1.1 - Number Formatting

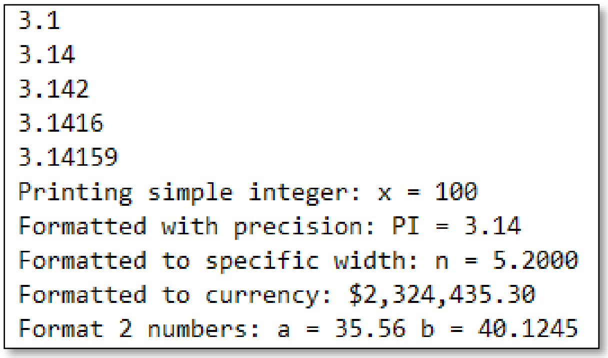
printf does not include a line return like **println** does. Insert \n at the end of a statement to get a new line.

Create a Java program named: **PrintfNumberFormatting.java**



```java
/**
 * Name: PrintfNumberFormatting.java
 * Written by:
 * Written on:
 * Purpose: How to print formated numbers in Java
 */

public class PrintfNumberFormatting {
    public static void main(String[] args) {
        // % - Replaced by variable
        // .1 - precision
        // f - floating point
        // \n - new line
        System.out.printf("%.1f\n", Math.PI);
        System.out.printf("%.2f\n", Math.PI);
        System.out.printf("%.3f\n", Math.PI);
        System.out.printf("%.4f\n", Math.PI);
        System.out.printf("%.5f\n", Math.PI);

        int x = 100;
        // d - integer
        System.out.printf(
                "Printing simple integer: x = %d\n", x);

        // this will print it up to 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);

      double currency = 2324435.3;
        // Number is formatted with commas
        System.out.printf(
                "Formatted to currency: $%,.2f\n", currency);

        double a = 35.55845;
        double b = 40.1245414;
        // printf with 2 variables
        System.out.printf("Format 2 numbers: a = %.2f b = %.4f", a, b);
    }
}
```

Example run:

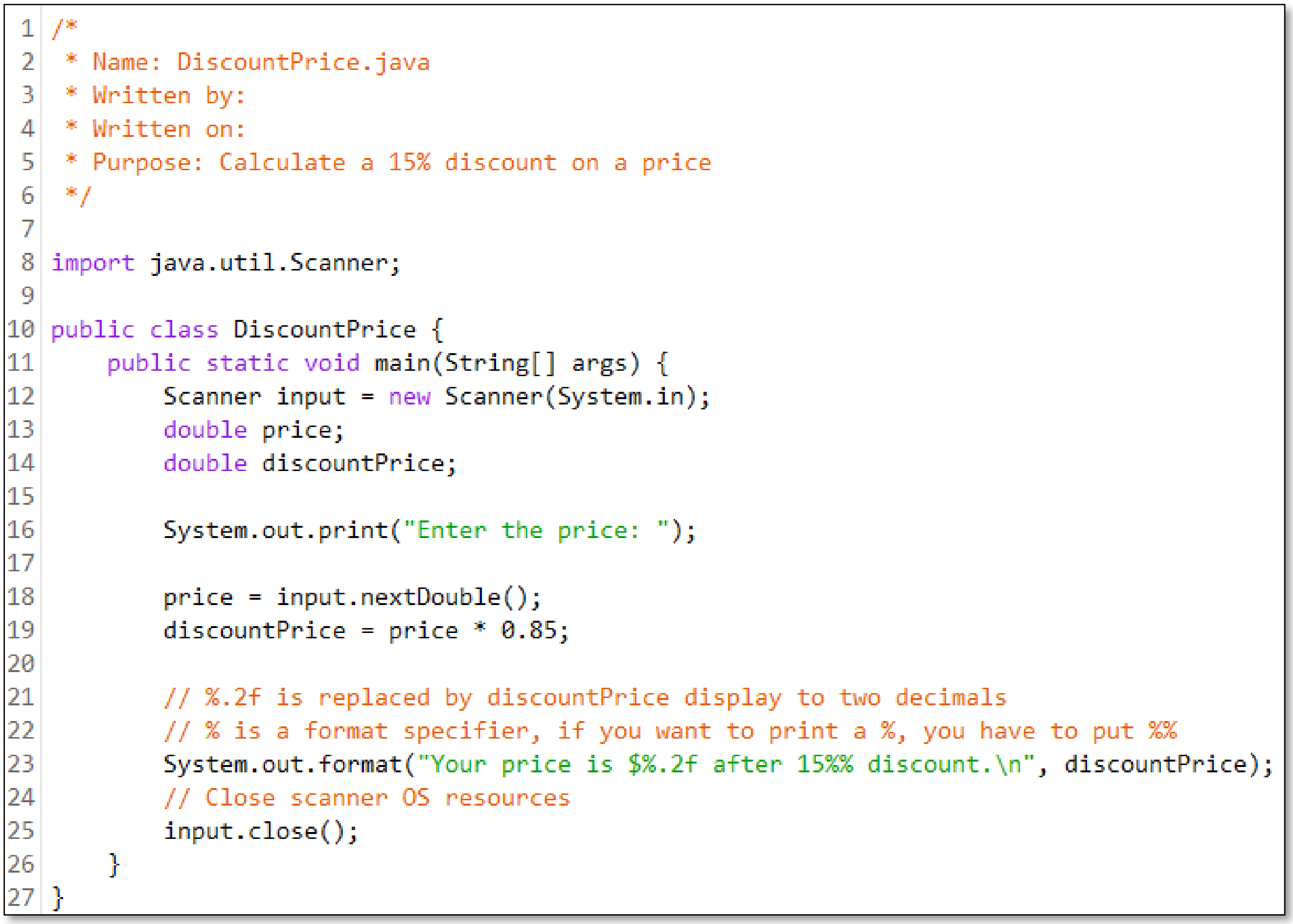


```
3.1
3.14
3.142
3.1416
3.14159
Printing simple integer: x = 100
Formatted with precision: PI = 3.14
Formatted to specific width: n = 5.2000
Formatted to currency: $2,324,435.30
Format 2 numbers: a = 35.56 b = 40.1245
```

Tutorial 1.2 - DiscountPrice with Format

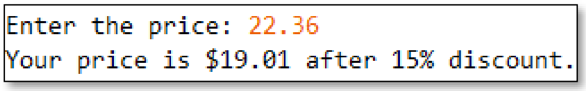
You can also use **System.out.format**, it behaves much the same as **printf**.

Create a Java program named **DiscountPrice.java**



```java
/*
 * Name: DiscountPrice.java
 * Written by:
 * Written on:
 * Purpose: Calculate a 15% discount on a price
 */

import java.util.Scanner;

public class DiscountPrice {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        double price;
        double discountPrice;

        System.out.print("Enter the price: ");

        price = input.nextDouble();
        discountPrice = price * 0.85;

        // %.2f is replaced by discountPrice display to two decimals
        // % is a format specifier, if you want to print a %, you have to put %%
        System.out.format("Your price is $%.2f after 15%% discount.\n", discountPrice);
        // Close scanner OS resources
        input.close();
    }
}
```

Example run:



```
Enter the price: 22.36
Your price is $19.01 after 15% discount.
```

Assignment Submission

1. Attach the pseudocode.
2. Attach the program files.
3. Attach screenshots showing the successful operation of the program.
4. Submit in Blackboard.