



# Numbers

By Dr. Subrat Kumar Nayak

Associate Professor

Department of CSE

ITER, SOADU

# Formatting Numbers

- **NumberFormat** is an **abstract base class** for all number formats. This class provides the interface for formatting and parsing numbers.
- NumberFormat also provides methods for determining which locales (US, India, Italy, etc) have number formats, and what their names are.
- NumberFormat helps you to format and parse numbers for any locale.
- A DecimalFormat object appropriate to the user's locale can be obtained from the factory method `NumberFormat.getInstance()` and manipulated using set methods.

## Package:

`Java.text.NumberFormat`

## How to Create a NumberFormat object?

`NumberFormat nf=NumberFormat.getInstance()` // No constructor

## Methods:

`setMaximumFractionDigits()`

- Sets the maximum number of digits allowed in the fraction portion of a number.

## Syntax:

`void setMaximumFractionDigits(int newValue)`

# Formatting Numbers

## Example:

```
import java.text.NumberFormat;

public class NumberFormatTest {

    public static void main(String args[])

    {

        NumberFormat nf=NumberFormat.getInstance();

        nf.setMaximumFractionDigits(2);

        double d=123.456;

        System.out.println(nf.format(d)); // format the value with format()

    }

}
```

## Output:

123.46

# Formatting Numbers

**void setMaximumIntegerDigits(int newValue)**

- Sets the maximum number of digits allowed in the integer portion of a number.

**void setMinimumFractionDigits(int newValue)**

- Sets the minimum number of digits allowed in the fraction portion of a number.

**void setMinimumIntegerDigits(int newValue)**

- Sets the minimum number of digits allowed in the integer portion of a number.

## **Example:**

Q. Write a program to set the minimum integer digit to 3, maximum fraction digit to 4 and minimum fraction digit to 2 of a decimal number.

# Formatting Numbers

## Changing the pattern dynamically

- You can also construct a `DecimalFormat` with a particular pattern or change the pattern dynamically using `applyPattern()`.

## Common Pattern Characters

Character	Meaning
#	A digit, leading zeroes are omitted.
0	A digit - always displayed, even if number has less digits (then 0 is displayed)
.	Locale-specific decimal separator (decimal point)
,	Locale-specific grouping separator (comma in English)
-	Locale-specific negative indicator (minus sign)
%	Shows the value as a percentage
;	Separates two formats: the first for positive and the second for negative values
'	Escapes one of the above characters so it appears
Anything else	Appears as itself

# Formatting Numbers

## Examples:

```
import java.text.NumberFormat;
import java.text.DecimalFormat;
public class NumberFormatTest {
    public static void main(String args[])
    {
        NumberFormat ourForm = new DecimalFormat("###.##");
        double d=123.345;
        System.out.println(ourForm.format(d));
    }
}
```

## Output:

123.34



# Formatting Numbers

```
import java.text.NumberFormat;
import java.text.DecimalFormat;
public class NumberFormatTest {
    public static void main(String args[])
    {
        NumberFormat ourForm = new DecimalFormat("0000.##");
        double d=12.5678;
        System.out.println(ourForm.format(d));
    }
}
```

Output:

0012.57

# Converting Between Binary, Octal, Decimal, and Hexadecimal

► `Integer.toString(int input, int radix)` to convert from integer to any type.

## Example

```
public class ConversionTest {  
    public static void main(String args[])  
    {  
        int i=42;  
        String res1=Integer.toString(i,2);  
        String res2=Integer.toString(i,8);  
        String res3=Integer.toString(i,16);  
        String res4=Integer.toString(i,10);  
        System.out.println("42 in base 2 is "+res1);  
        System.out.println("42 in base 8 is "+res2);  
        System.out.println("42 in base 16 is "+res3);  
        System.out.println("42 in base 10 is "+res4);  
    }  
}
```

Output:

42 in base 2 is 101010  
42 in base 8 is 52  
42 in base 16 is 2a  
42 in base 10 is 42



# Converting Between Binary, Octal, Decimal, and Hexadecimal

- Integer.parseInt(String input, int radix) to convert from any type of number to an Integer

```
public class ConversionTest {  
    public static void main(String args[])  
    {  
        String str="1010";  
        Integer iObj=Integer.parseInt(str,2);  
        System.out.println("1010 in base 2 is "+iObj);  
        Integer iObj1=Integer.parseInt(str,8);  
        System.out.println("1010 in base 8 is "+iObj1);  
        Integer iObj2=Integer.parseInt(str,16);  
        System.out.println("1010 in base 16 is "+iObj2);  
    }  
}
```

Output:

1010 in base 2 is 10

1010 in base 8 is 520

1010 in base 16 is 4112



End of Session