# java.text.DecimalFormat

The API class **`java.text.DecimalFormat`** allows you to convert a number into a string that represents the number. You use a "pattern" to describe how you want your number to appear. The pattern uses special symbols to represent characters within the string. Here are some of the special symbols that you can use.

| A Short List of Pattern Symbols | |
|---|---|
| **Symbol** | **Meaning** |
| **`0`** | Digit, leading & trailing zeros appear |
| **`#`** | Digit, leading & trailing zeros suppressed |
| **`-`** | Minus sign |
| **`.`** | Decimal point |
| **`,`** | Grouping separator (e.g. 1,000) |

***Examples***

| *Pattern* | *Number* | *Resulting String* |
|---|---|---|
| **`"00.00"`** | 5.5 | **`05.50`** |
| **`"##.##"`** | 5.5 | **`5.5`** |
| **`"##.00"`** | 5.5 | **`5.50`** |
| **`"#,###.00"`** | 12345.678 | **`12,345.68`** |

Here's one of the constructors to the class:

```
public DecimalFormat( String pattern )
// Build a DecimalFormat object using the pattern passed
// as an argument.
```

***Example***

This Java statement uses the constructor to build a **`DecimalFormat`** object.

```
DecimalFormat df = new DecimalFormat( "##.00" );
```

Once you've built the object, you can call its **format** method to convert a number into a string; format is overloaded to accept both **double** and **long** values.

```
public String format( double number )
// Convert the double argument into a string.
```

```
public String format( long number )
// Convert the long argument into a string.
```

***Example***

This Java code fragment outputs the value **5.50**.

```
DecimalFormat df = new DecimalFormat( "##.00" );
double x = 5.5;
System.out.println( df.format( x ) );
```

***Example***

This Java code fragment outputs the value **$1,234,567.00**.

```
DecimalFormat df = new DecimalFormat( "$#,###.00" );
[redacted] int m = 1234567;
System.out.println( df.format( m ) );
```

## Exercises

Give the output of each code segment.

1.
```
int a = 2;
DecimalFormat dfa = new DecimalFormat( "##" );
System.out.printf( dfa.format( a ) );
```

2.
```
int b = 2;
DecimalFormat dfb = new DecimalFormat( "00" );
System.out.printf( dfb.format( b ) );
```

3.
```
int c = 2;
DecimalFormat dfc = new DecimalFormat( "-##" );
System.out.printf( dfc.format( c ) );
```

Give the output of each code segment.

4.
```
int d = 2000000;
DecimalFormat dfd = new DecimalFormat( "#,###" );
System.out.printf( dfd.format( d ) );
```

5.
```
int e = 2000000;
DecimalFormat dfe = new DecimalFormat( "#,###.##" );
System.out.printf( dfe.format( e ) );
```

6.
```
int f = 2000000;
DecimalFormat dff = new DecimalFormat( "#,###.00" );
System.out.printf( dff.format( f ) );
```

7.
```
double g = 3.54;
DecimalFormat dfg = new DecimalFormat( "#" );
System.out.printf( dfg.format( g ) );
```

8.
```
double h = 3.54;
DecimalFormat dfh = new DecimalFormat( "#.#" );
System.out.printf( dfh.format( h ) );
```

9.
```
double i = 3.54;
DecimalFormat dfi = new DecimalFormat( "#.###" );
System.out.printf( dfi.format( i ) );
```

10.
```
double j = 3.54;
DecimalFormat dfj = new DecimalFormat( "#.000" );
System.out.printf( dfj.format( j ) );
```

11. A real estate program keeps house prices in the variable **`price`**. House values range up to $999,999. Write the Java statements that use a **`DecimalFormat`** object to correctly format and print **`price`**.

12. A payroll program keeps a worker's hourly wage in the variable **`wage`**. Wages range up to $99.99. Write the Java statements that use a **`DecimalFormat`** object to correctly format and print **`wage`**.

13. A student records program keeps a student's grade point average in the variable **`gpa`**. Grade point averages range from 0.0 to 4.0 and are kept to 3 decimal places (e.g. 3.125). Write the Java statements that use a **`DecimalFormat`** object to correctly format and print **`gpa`**.