Floating-point Numbers

Unlike whole numbers, floating-point numbers have fractional parts that we express with a decimal point.

In this table, you can see some examples of both whole numbers and floating point numbers, in comparison.

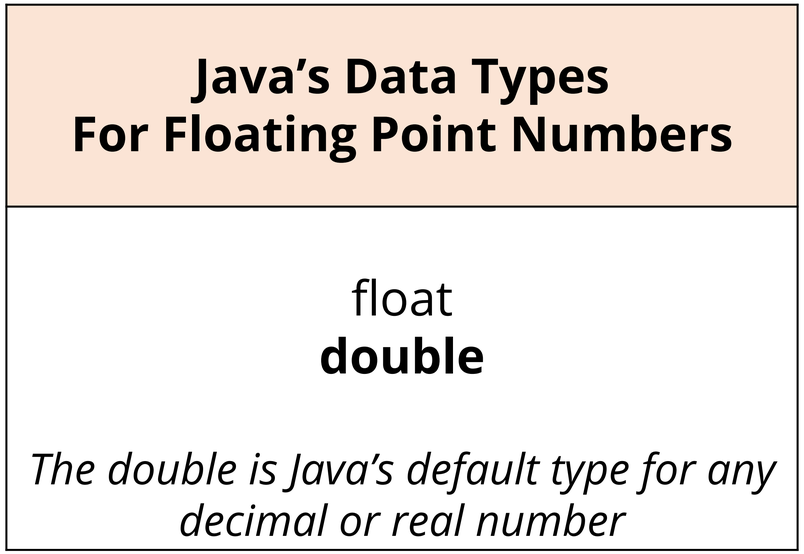
Whole number Examples	Floating Point Examples
3	3.14159
100000	10.0
-2147483649L	-0.6666666666666667

Floating-point numbers are also known as real numbers.

Floating-point Number Data Types

We use a floating-point number when we need more precision in calculations.

There are two primitive types in Java for expressing floating-point numbers, the float and the double.



The double is Java's default type for any decimal or real number.

Single and Double Precision

Precision refers to the format and amount of space occupied by the relevant type.

This table shows the width of each of the floating point types and their ranges.

The ranges are shown in Java's scientific notation, which we show below in blue color.

Data Type	Width (in bits)	Min Value	Max Value
float	32	1.4E-45	3.4028235E38
double	64	4.9E-324	1.79769313486231 57E308

You can see the e-notation followed by either a positive or negative number.

Java's Scientific Notation

Scientific notation can be translated into more familiar terms, by replacing the 'E' in the number, with the phrase 'times 10 to the power of'.

1.4E-45 is the same as 1.4×10^{-45} and 3.4E38 is the same as 3.4×10^{38}

Data Type	Min Value	Max Value
float	1.4E-45	3.4028235E38

So we can say, the minimum value of a float is $1.4 * 10^{-45}$ and its maximum value is approximately $3.4 * 10^{38}$.

Java's Scientific Notation

Think about that for a moment, using the double's minimum value shown below, remembering that 10-1 = 0.1 and 10-5 = 0.00001 for example.

Imagine writing out the double data type's minimum value in decimal format! That would be a lot of zeros after the decimal.

Data Type	Min Value	Max Value
double	4.9E-324	1.79769313486231 57E308

I hope you can see that a double, when compared to a float, can represent both a much smaller decimal value, and a much larger decimal value. This is why its called more precise.

Because it's more precise, the double is the default type for floating point numbers.

float and double and numeric literal suffixes

Important: The double data type is Java's default type for real numbers.

- For example, any number with a decimal is a double.
- So, 10.5 is a double by default in Java.

- The double data type can be specified as a numeric literal with a suffix of either lowercase 'd', or uppercase 'D', but because doubles are the default in Java, the suffix is optional to use.
- On the other hand, the float data type can be specified as a numeric literal with a suffix of lowercase 'f', or uppercase 'F'. This suffix is required if you are assigning a real number to a variable that was declared with a float type.

Challenge

Thinking back to our discussion on casting, how do you think you'd do the same for the float to remove this error? I am talking about using casting here, specifically, because, as you have learned, we could just use the suffix f to tell Java this is a float. Here I want you to use casting.

```
jshell> float myOtherFloatValue = 5.25;
|  Error:
|  incompatible types: possible lossy conversion from double to float
|  float myOtherFloatValue = 5.25;
|                            ^--^
```

Certification Exam Pointer

Not everyone realizes that Java's default data type for a decimal literal is a double, which is larger and more precise than a float.

Java likes to put a similar line of code in its code segments on exam questions, to what we saw earlier, omitting that 'f' suffix. Without a computer to check, this statement can look fairly innocuous.

```
float myOtherFloatValue = 5.25;
```

The number 5.25 is a double, so assigning it to a float will raise an error.

This is a gift question to an exam taker, if you can easily spot this compiler error.