

Primitive Types

Floating Point Numbers
(double)

Lecture Contents

- Review of Java Integers (`int`)
- How to Store a Binary Floating Point Number
- Java `double` type

Review of Java Integers (`int`)

- Modern computers organize **bits** into **bytes** (8 bits = 1 byte).
- Java programmers can choose from the following **types** for storing integers

type	bits	numerical range
byte	8	-128 to +127
short	16	-32768 to +32767
int	32	-2,147,483,648 to +2,147,483,647
long	64	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807

- For this course, unless you have a specific reason, always use the type **`int`** to store integers.

Lecture Contents

- Review of Java Integers (`int`)
- **How to Store a Binary Floating Point Number**
- Java `double` type

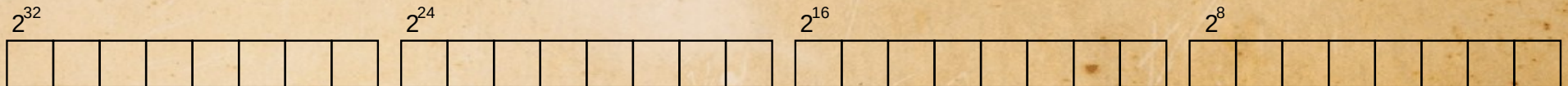
How to Store a Binary Floating Point Number?

- What about numbers with decimals (floating point numbers)?

$$\pi = 3.1415926535897932384626433\dots$$

- What about huge numbers?
 - Avogadro's Number:

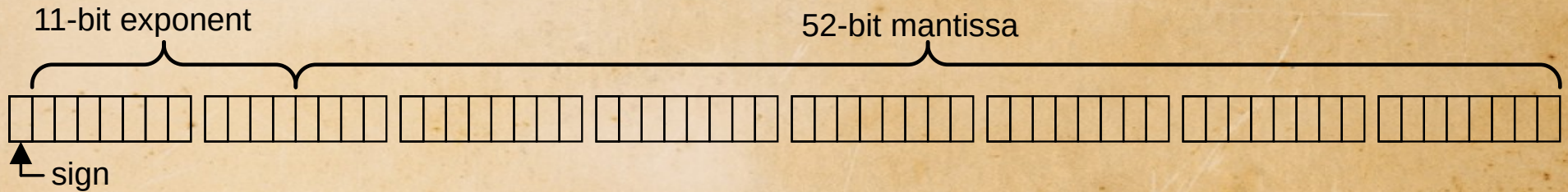
$$N_A = 6.02214076 \times 10^{23}$$



How to Store a Binary Floating Point Number?

- Floating point numbers
 - Scientific notation

$$\pm \textit{mantissa} \times 2^{\textit{exponent}}$$



How to store numbers?

- Java floating point numbers

	sign	exponent	mantissa
float	1	8	23
double	1	11	52

- Specification: *IEEE 754*

Using Floating Point Numbers

```
System.out.println("Hello " + 3.0);
```

```
System.out.println("Hello " + 3.0 + 5.0);
```

```
System.out.println("Hello " + (3.0 + 5.0) );
```

```
System.out.println(3 + 5.0);
```

```
System.out.println(3.0 + 5);
```

Using Floating Point Numbers

```
System.out.println("Hello " + 3.0);
```

```
System.out.println("Hello " + 3.0 + 5.0);
```

```
System.out.println("Hello " + (3.0 + 5.0) );
```

```
System.out.println(3 + 5.0);
```

```
System.out.println(3.0 + 5);
```

Hello 3.0

Hello 3.05.0

Hello 8.0

8.0

8.0

Using Floating Point Numbers

```
System.out.println(3.0 / 5.0);
```

```
System.out.println(3 / 5.0);
```

```
System.out.println(3.0 / 5);
```

```
System.out.println(3 / 5);
```

Using Floating Point Numbers

```
System.out.println(3.0 / 5.0);
```

```
System.out.println(3 / 5.0);
```

```
System.out.println(3.0 / 5);
```

```
System.out.println(3 / 5);
```

0.6

0.6

0.6

0

Primitive Types

Floating Point Numbers
(double)