

Primitive Types

Floating Point Numbers
(double)

Lecture Contents

- List of Java Primitive Types
- Review of Java Integers (`int`)
- How Computers Store Binary Floating Point Numbers
- Using Floating Point Numbers (`double`)

List of Java Primitive Types

- *Integers*
 - byte (8 bits)
 - short (16 bits)
 - **int** (32 bits)
 - long (64 bits)
- *Real Numbers*
 - float (32 bits)
 - **double** (64 bits)
- *True/False*
 - **boolean** (1 bit?)
- *Letters*
 - char (16 bits)



This
Lecture

Note: Primitive types tested in
AP Computer Science A
are given in bold font.

Lecture Contents

- List of Java Primitive Types
- **Review of Java Integers (`int`)**
- How Computers Store Binary Floating Point Numbers
- Using Floating Point Numbers (`double`)

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

- List of Java Primitive Types
- Review of Java Integers (`int`)
- **How Computers Store Binary Floating Point Numbers**
- Using Floating Point Numbers (`double`)

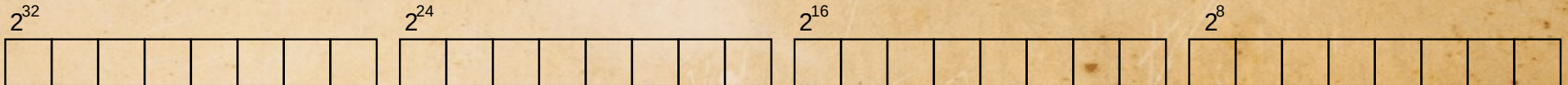
How Computers Store Binary Floating Point Numbers

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

$$\pi = 3.1415926535897932384626433\dots$$

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

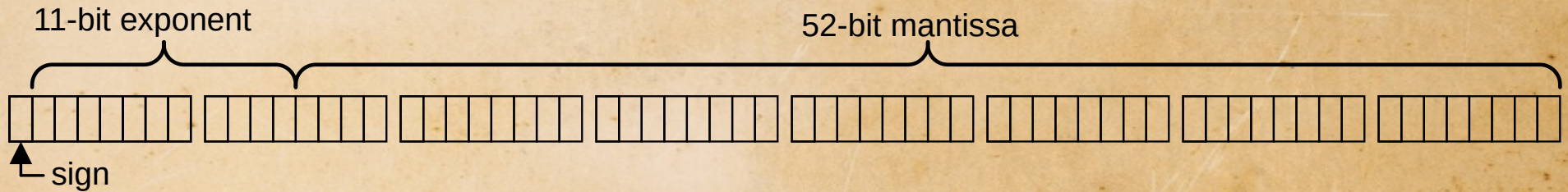
$$N_A = 6.02214076 \times 10^{23}$$



How Computers Store Binary Floating Point Numbers

- Floating point numbers
 - Similar to scientific notation

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



How Computers Store Binary Floating Point Numbers

- Java floating point numbers

	sign	exponent	mantissa
float	1	8	23
double	1	11	52

- Specification: *IEEE 754*

Lecture Contents

- List of Java Primitive Types
- Review of Java Integers (`int`)
- How Computers Store Binary Floating Point Numbers
- **Using Floating Point Numbers (`double`)**

Using Floating Point Numbers (double)

```
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 (double)

```
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

The + operator will concatenate strings and add numbers.

Using Floating Point Numbers (double)

- The + operator will *concatenate* strings and add numbers.
- When a string and floating point number (double) are operands to a + operation, the number is converted to a character string and then concatenated to the string.
 - `System.out.println("Hello " + 3.0);` → `Hello 3.0`
- The + operations are performed from left to right.
 - `System.out.println("Hello " + 3.0 + 5.0);` → `Hello 3.05.0`

Using Floating Point Numbers (double)

```
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 (double)

```
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

Using Floating Point Numbers (double)

- When two integers (`int`) are in a calculation, the result is an integer. The number is *truncated* at the decimal point, removing any values after it (no rounding of the number).
 - `System.out.println(11 / 4);` → 2
- When an integer (`int`) and a floating point number (`double`) are in a calculation, the integer is converted to a floating point number, then the operation is performed. (There is no loss of precision.)
 - `System.out.println(3 / 5.0);` → 0.6
 - `System.out.println(3.0 / 5);` → 0.6

List of Java Primitive Types

- *Integers*
 - byte (8 bits)
 - short (16 bits)
 - **int** (32 bits)
 - long (64 bits)
- *Real Numbers*
 - float (32 bits)
 - **double** (64 bits)
- *True/False*
 - **boolean** (1 bit?)
- *Letters*
 - char (16 bits)



This
Lecture

Note: Primitive types tested in
AP Computer Science A
are given in bold font.

Primitive Types

Floating Point Numbers
(double)