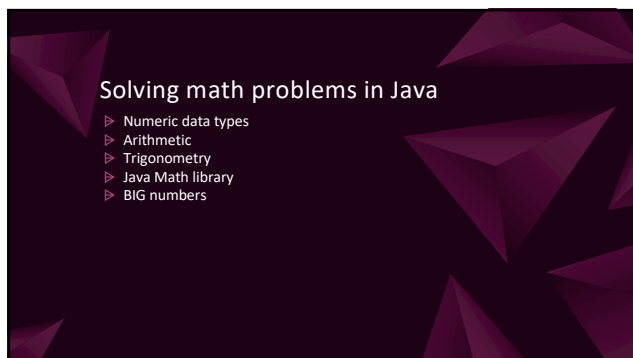
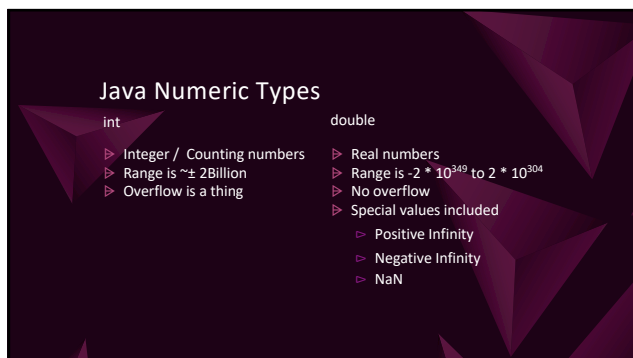




1



2



3

PEMDAS

- Parens are first and group statements only – no multiplication!
- Exponents are handled with Math.pow, no special operator
- Multiplication is done with the * operator
- Modulo with the % operator
 - You do long division, ignore the answer and keep the remainder
- Division is done with the / operator
 - Data type matters!!!!
- Addition is with the + operator
- Subtraction is with the - operator

4

Comparing numeric values

- | int | double |
|----------------------------|--|
| ‣ Compare equality with == | ‣ Do NOT use == |
| ‣ Inequality with != | ‣ Subtract two doubles and compare the difference against some predetermined close enough amount |
| | ‣ Use Math.abs to make it easy |

5

Casting

- Changing values from double to int requires an explicit cast as the extra information contained in a double has no where to go in an int.
- Casting up to a double should be done BEFORE the mathematical operation(s) so information is not lost
 - `double demo = (double) 7 / 4;`
- Casting down to an int should be done AFTER the operation(s) for the same reason
 - `int otherDemo = (int) (Math.random() * 100);`

6

Random

- ▶ `Math.random()` is the default way to get random numbers in Java
- ▶ It returns a double value in the range `[0,1)`
 - ▷ AKA it can include 0 but NOT 1
- ▶ I use it most often by multiplying by some scalar and then casting as an int.
- ▶ You can also shift the range up/down the number line by adding or subtracting after multiplying
- ▶ Use `.size()` or `.length` as the scalar for a valid random index
- ▶

```
int randomValue = (int) (Math.random() * scalar) + shift;
```

7

Must know Math methods

- ▶ `Math.abs`
 - ▷ Absolute value
 - ▷ Absolutely helpful
- ▶ `Math.max` / `Math.min`
 - ▷ Overloaded for int and double
 - ▷ Returns the bigger/smaller of the two parameters
 - ▷ Very nestable!

8

More in Math

- ▶ `Math.pow`
 - ▷ How to do exponents
 - ▷ Handles positive/negative and fractional (root) values!
- ▶ `Math.sqrt`
 - ▷ Square roots minus the vowels

9

Trigonometric functions

- ▶ These all use radian by default so remember your unit circle!
- ▶ Take and return double values
- ▶ One of the few useful constants in Math.PI

Standard	Inverse
<code>sin(angle)</code>	<code>asin(angle)</code>
<code>cos(angle)</code>	<code>acos(angle)</code>
<code>tan(angle)</code>	<code>atan(angle)</code>
<code>atan2(y,x)</code>	

10

Rounding / Bounds

- ▶ The methods `round`, `floor`, and `ceil` return values based only on the fractional part of the double value.
- ▶ They return long as the data type so will need to be downcast in to an `int`
- ▶ `Math.round` behaves the way you learned about rounding fractions in elementary
- ▶ `Math.floor` ALWAYS drops all fractional value, regardless of how large the fractional portion is.
- ▶ `Math.ceil` ALWAYS goes to the next whole value regardless of how small the fractional portion is.

11

Logarithms

- ▶ If you plan on writing any code to help with calculus you will be using `Math.log` and `Math.log10`
- ▶ `Math.log` is the natural logarithm of the supplied parameter
- ▶ `Math.log10` is the base 10 logarithm (quelle surprise)

12

Big Numbers

- ▶ What if you want to do math on arbitrarily LARGE values and your want precise answers?
 - ▶ Integers: `java.math.BigInteger`
 - ▶ Decimal values: `java.math.BigDecimal`
- ▶ Slower than `int`/`double` but does not have the overflow problem of `int` or the lack of precision of a `double`.

13

Other methods

- ▶ There are other methods in Math I just did not cover everything. Check out the Java Math API for even more information

14
