Panel 1

**Last Line**

- Declaring variables
- Variable names
- Assigning values
- Basic operators: `+`, `/`, `//`, and `%`

- $x \% y$ is remainder after dividing $x$ by $y$ as int.

**Example:**

- $5 \% 2 = 1$
- $15 \% 4 = 3$

Panel 2

**Homework**

1. What does the `%` operator do? ✓

2. What is the value of $z$:

   ```
   int x = 5;
   int y = 2;
   int z = y / x;  //
   ```

   
   Explain!

   **Rule:** if both arguments are int, answer is int (??)

   if one (or both) are double, answer is double (good)
Panel 3

Suppose we have variables declared and initialized as follows:

```c
int i1 = 10, i2 = 3;
double x1 = 8.0, x2 = 5.0;
```

What are the resulting values and types of the computations:

- `i1 * i2;` 30
- `x1 / x2;` 1.6
- `i1 / i2;` 3
- `i1 - x1;` 2.0
- `i1 % i2;` 1
- `i1 & x1;` 2.0

Panel 4

_Short cut Notation_

Frequently want to: "add 5 to x"

\[ x = x + 5 \] is not equation, because = is assigned

If \( x = 2 \) and I want \( x = x + 5 \) then \( x \to 7 \)

_Hint_

\[ \sqrt{x + 5} \] or \( x = x + 5 \)

So

\[ x = x - 5 \]

Worse: \( x++ \to x = x+1 \) (increment)

\( x-- \to x = x-1 \) (decrement)
Panel 5

What is the value of $z$ at the end of this code fragment?

1. $\text{int } z = 23;$
2. $\text{int } x = 8, y = 53;$
3. $x *= y; \quad x = x \cdot y = 8 \cdot 53 = 374$
4. $y /= 2; \quad y = y/z = 53/2 = 26$
5. $x++; \quad x = x+1$
6. $z += (x + y); \quad z = z + (x+y)$

$$x = 8 \quad y = 53 \quad z = 374 + 374 + 374 = 1122$$

$$y = 53$$

$$z = 23$$

Panel 6

Advanced Arithmetic for Numeric Types

To use a mathematical function, prefix it by the word `Math`, followed by a dot, and the function's name. Each function returns the type double unless otherwise indicated.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>abs(number)</td>
<td>Absolute value of a number; returns same type as input value</td>
</tr>
<tr>
<td>min(a, b)</td>
<td>Greater value of $a$ or $b$, returns same type as input value</td>
</tr>
<tr>
<td>minf(max, min)</td>
<td>The number with the greater value, raised to the power of $a$</td>
</tr>
<tr>
<td>sqrt(number)</td>
<td>Square root of a number</td>
</tr>
<tr>
<td>random()</td>
<td>A random number between 0.0 and 1.0</td>
</tr>
<tr>
<td>ceil(number)</td>
<td>Smallest (closest to positive infinity) value that is not less than number and is equal to a mathematical integer</td>
</tr>
<tr>
<td>floor(number)</td>
<td>Largest (closest to negative infinity) value that is greater than number and is equal to a mathematical integer</td>
</tr>
<tr>
<td>round(number)</td>
<td>The value of a number rounded to the nearest integer value. Returns <code>int</code> if <code>number</code> is an <code>int</code>, and <code>long</code> if <code>number</code> is a <code>double</code></td>
</tr>
</tbody>
</table>

Mathematical Constants

- `Math.E`: The value that is closer than any other to $e$, the base of the natural logarithm.
- `Math.PI`: The value that is closer than any other to $\pi$, the ratio of the circumference of a circle to its diameter.
- `Double.POSITIVE_INFINITY`: A representation of positive infinity of type double.
- `Double.NEGATIVE_INFINITY`: A representation of negative infinity of type double.
- `Double.NaN`: A representation of "Not-a-Number".
Panel 7

Example:
Write some Java code segments to compute:

1. The circumference and area of a circle with radius 3.0.

2. The length of the hypotenuse of a right triangle given its base and height.

3. The angle in degrees, given that \( \text{rangle} \) is the angle in radians.

4. The angle in radians, given that \( \text{dangle} \) is the angle in degrees.

5. What is \( \exp(4000.0) \)? What is \( 10.0/0.0 \)? What is \( -10.0/0.0 \)? What is \( 0.0/0.0 \)?

Panel 8

Logical Operators and Examples

Logical variables can not be added, subtracted, etc.

Instead:

- and \( \& \& \)
- or \( \| \)
- not \( ! \)

Comparison Operators

- \(<, \leq, , >, \geq, =, !=\)

- \(==\) in equal

- \(!=\) not equal
Recall logic tables:

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>x &amp; y</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

| x | y | x || y |
|---|---|-----|
| T | T | T   |
| T | F | T   |
| F | T | T   |
| F | F | F   |

Panel 9

Panel 10

1. Determine if the following statements are true or false, using:

   int x = 5, y = 9, z = 2;
   boolean r = false, result;

   a. result = ( 5 >= x );  true

   b. result = ( y < z );  false

   c. result = (y == 9) && (!(x >= 2) || r );

   c. result = !( ( x || ( y < x ) ) && ( 5 >= x ) );

   c. result = ( x + y ) >= ( 16 - z ) ;

   c. result = ( x == ( z++) + 2 ) ;

   c. result = ( x == (++z) + 2 ) ;

   c. result = ( !x && ( -(y) > (z + x) ) );

   c. result = (((21 - z) == (((y++) + x + z) + 1)) && (!r == true));