cast operations:

Note: char is actually also a numeric type.

Ascii:

\[
\text{A} \leftrightarrow 65 \leftrightarrow 01000001
\]

Unicode:

\[
\text{'A'} \leftrightarrow 65 \leftrightarrow \text{00000000 01000001}
\]
Ex. CharCode1.java

CharCode2.java

more on directives:

Assignment

variable = expression;

Ex.

```java
int x, y, z;
z = 5;
x = (y = z);
equivalent:
x = y = z;
```
Variations on assignment

Compound assignments:

\[ a += b \quad \text{same as} \quad a = a + b \]
\[ a -= b \quad \ldots \quad a = a - b \]
\[ a *= b \quad \ldots \quad a = a * b \]
\[ a /= b \quad \ldots \quad a = a / b \]
\[ a %= b \quad \ldots \quad a = a \% b \]

\[ a = a / b \] only for integer types.

\[ a = a / b \] different for f.p.

\[ a \% b \] vs. integer
Ex.

\[ 14.0 / 5.0 \text{ eval to } 2.8 \]

\[ 14 / 5 \quad \text{ is } \quad 2 \]

\[ 14 \% 5 \quad \text{ is } \quad 4 \]

Let a be dividend, b divisor.

Both integers.

\[ a = (a/b) \times b + (a \% b) \]

\[ 0 \leq (a \% b) < b \]
Ex. `HMS.java`

more shortcuts:

<table>
<thead>
<tr>
<th>Postfix</th>
<th>Prefix</th>
<th>Same as</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto-increment</td>
<td><code>a++</code></td>
<td><code>++a</code></td>
</tr>
<tr>
<td>auto-decrement</td>
<td><code>a--</code></td>
<td><code>--a</code></td>
</tr>
</tbody>
</table>

The difference is its value as an expression:

- `(a++)` evaluates to value of `a` before increment, old value
- `(++a)` evaluates to value of `a` after increment, new value
Ex. AutoIncrementDecrement.java

```
i = a - b - c + d - e + 1
i = i + 7 + 7 - 13 - 3
i = i + 8 + 6 + 1
```

Operator Precedence

```
a op1 b op2 c
```

Does this mean:

- `(a op1 b) op2 c` if `op1` >= `op2`

or

- `a op1 (b op2 c)` if `op1` < `op2`
Ternary Conditional Operator

? : :

cond ? expr1 : expr2

boolean expression

evaluated if cond is true
evaluated if cond is false

Ex. Plural.java