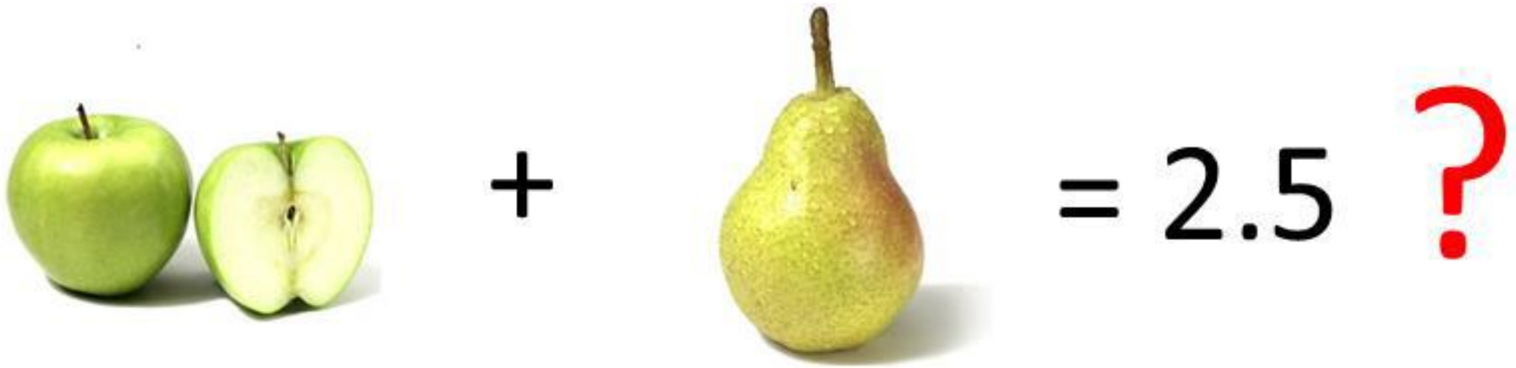
User-Defined Literals



Syntax: <built_in-Literal> + _ + <Suffix>

- Natural numbers: `10101010_b`
- Floating-point numbers: `123.45_km`
- C-string literals: `"hello"_i18n`
- Character literals: `'1'_character`

User-Defined Literals

The C++ runtime maps the user-defined literal on the corresponding literal operator

- `1_m` ➡ `operator "" _m(1){ ...`
- `"hello"_i18n` ➡ `operator "" _i18n("hello", 5)`

- The literal operator is available.
  - cooked and raw form for natural numbers and floating-point numbers
  - raw for C string literals and character literals

- The cooked form has higher priority.

User-Defined Literals

- Cooked form
  - Accepts its arguments as `long double` or `unsigend long long int`

    `1.45_km` ➡ `operator "" _km(1.45)`

- Raw form
  - Accepts its arguments as `(const char*, size_t)`, `(const char*)` or `const char`

    `1.45_km` ➡ `operator "" _km("1.45")`

- Syntax
  - Between "" and `_km` must be an empty space.
  - Literals should start with underscore `_km` to distinguish them from the built_in literals.

`userDefinedLiteral.cpp`