Expressions

- Composed of operators and operands
- Evaluate into a value

Some Operators:

- Unary
  - +A
  - –A
  - ++A
  - --A
  - A++
  - A--
- Binary
  - A+B
  - A-B
  - A*B
  - A/B
  - A%B
  - A = B

Precedence and Associativity

- Precedence
  - Which comes first
- Associativity
  - Left to right or right to left
- Group the operator with the operands
- But do not enforce execution order completely

Short Circuit

- Enforces order of evaluation between LHS and RHS
- Compiler dependency
  - E.g. if(--j || i++ & --j)
    - LHS first, RHS MAYBE
    - Only apply to || and &&
- Expr1 && expr2
  - If exp1 evaluate to 0, exp2 is not done
  - If exp1 evaluate to 1, exp2 is then evaluate
- Expr1 || expr2

    - If exp1 evaluate to 1, exp2 is not done
    - If exp1 evaluate to 0, exp2 is then evaluate

<u>Ternary operator</u>

if(exp1) exp2
else exp3

Rewrite to: exp1 ? exp2 : exp3

exp1 != 0 result is exp2 else exp3

<u>Precedence</u>

- Setting the priority of operation
- ++i , --i (pre increment)
    - Evaluate to the increment value of i
- i++ , i-- (post-increment)
    - Evaluate to the original value of i
    - Then increment itself

<u>L-Value</u>

- Refers to objects with a memory location

<u>R-Value</u>

- Temporary values
- Every C operator evaluate to an R value
- Will be discarded if not assign regardless