Introduction to Expressions

As part of Week 3, Unit 3

CC4 Data Structures and Algorithms

Lovely Jenn A. Reformado

Table of Contents

- Introduction to expressions
- Infix expressions
- Importance of prefix and postfix expressions
- Postfix expressions
- Prefix expressions

College of
Information Technology
and Computer Science
CENTER OF EXCELLENCE
in Information Technology

Introduction to Expressions

What are expressions? | Parts of an expression | Expressions in this unit

What are Expressions?

- A combination of symbols used to express portions of mathematical equations
- Symbols can be:
  - Numbers (positive / negative, absolute / decimal)
  - Variables (expressed in letters)
  - Operators (arithmetic or comparison)
  - Punctuations (groupings)
- Mathematical expressions can be:
  - Arithmetic (numbers only)
  - Algebraic (numbers and constants)
  - Comparison (uses comparison symbols)
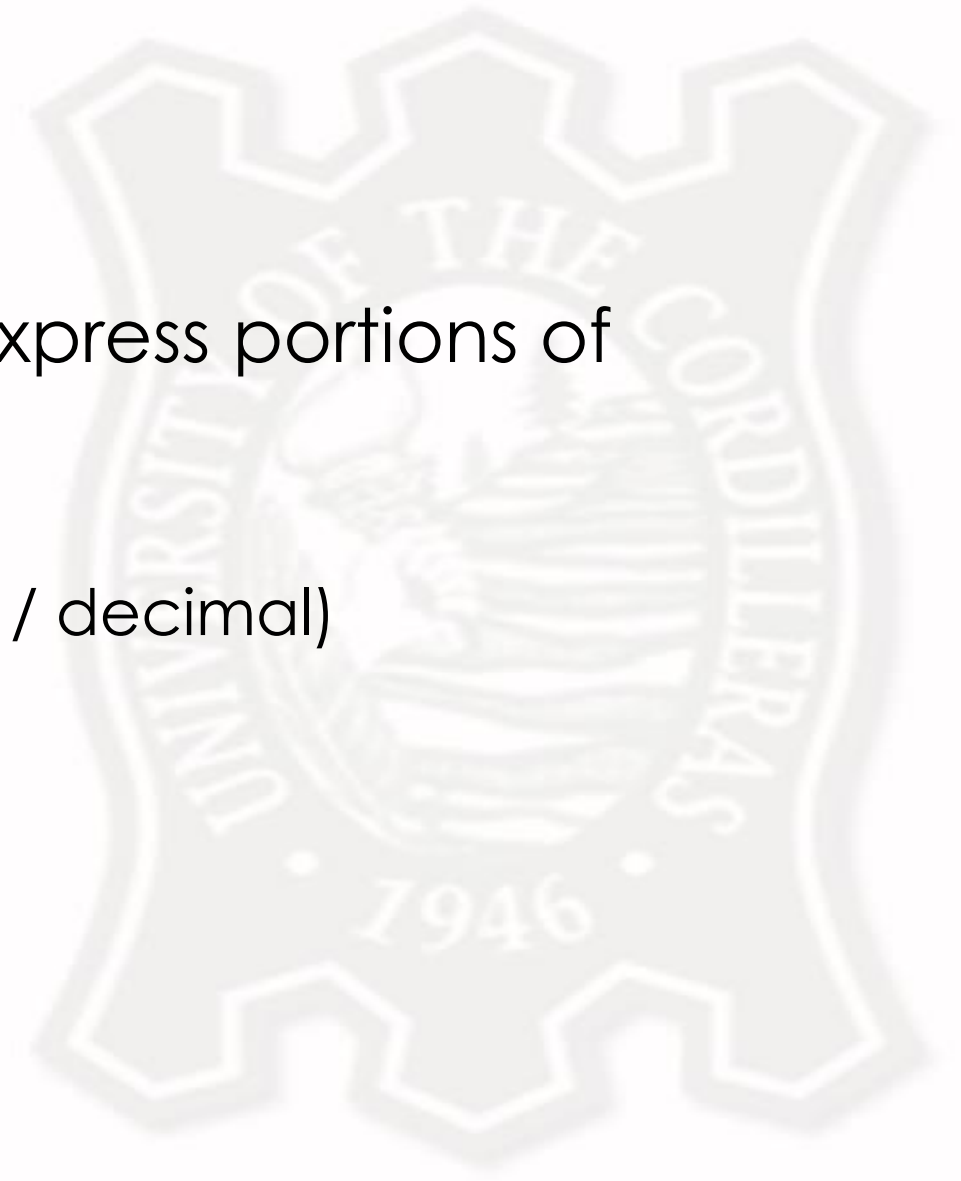
Parts of an Expression

Operator

- Symbol(s) that decide(s) which operation is to be performed
- Arithmetic (+, -, *, /, %)
- Comparison (<, >, ==, <=, >=)

Operand

- Symbol(s) that represents an entity on which the operation is performed
- Numbers (0, -1, 1.2)
- Variables (a, b, x)
- Constants (log, e)

Expressions in this Unit

- Limited to binary expressions
  - Utilizes operations that would require two operands
  - Arithmetic and comparison operations may be used
  - Increment and decrement operations are not to be used
  - Unary / ternary operators are possible for use in expressions
- Limited to arithmetic operations
  - Comparison operations may be used, and may appear in activities
  - All examples will be limited to arithmetic operations and the equal sign (=)

Infix Expressions

What is an infix expression? | Examples of infix expressions | Precedence and associativity | Importance of prefix and postfix expressions

What is an Infix Expression?

- Usual way on how humans express mathematical expressions
- Notation where the operators are placed in-between the operands
- Requires specific precedence and associativity rules

Examples of Infix Expressions

- a + b

- c + d * 2

- 3 – e + 1

- (x + y) * (z – 5)

Precedence and Associativity

Precedence

- Determines the order of what operation must be performed first in comparison to other operations
- Parenthesis, Multiplication and Division, Addition and Subtraction
- Example: c + d * 2, (x + y) * (z – 5)

College of
Information Technology
and Computer Science
CENTER OF EXCELLENCE
in Information Technology

Precedence and Associativity

Associativity

- Determines the order of what operation must be performed if they are in the same precedence
- Left to right precedence
- Example: 3 – e + 1
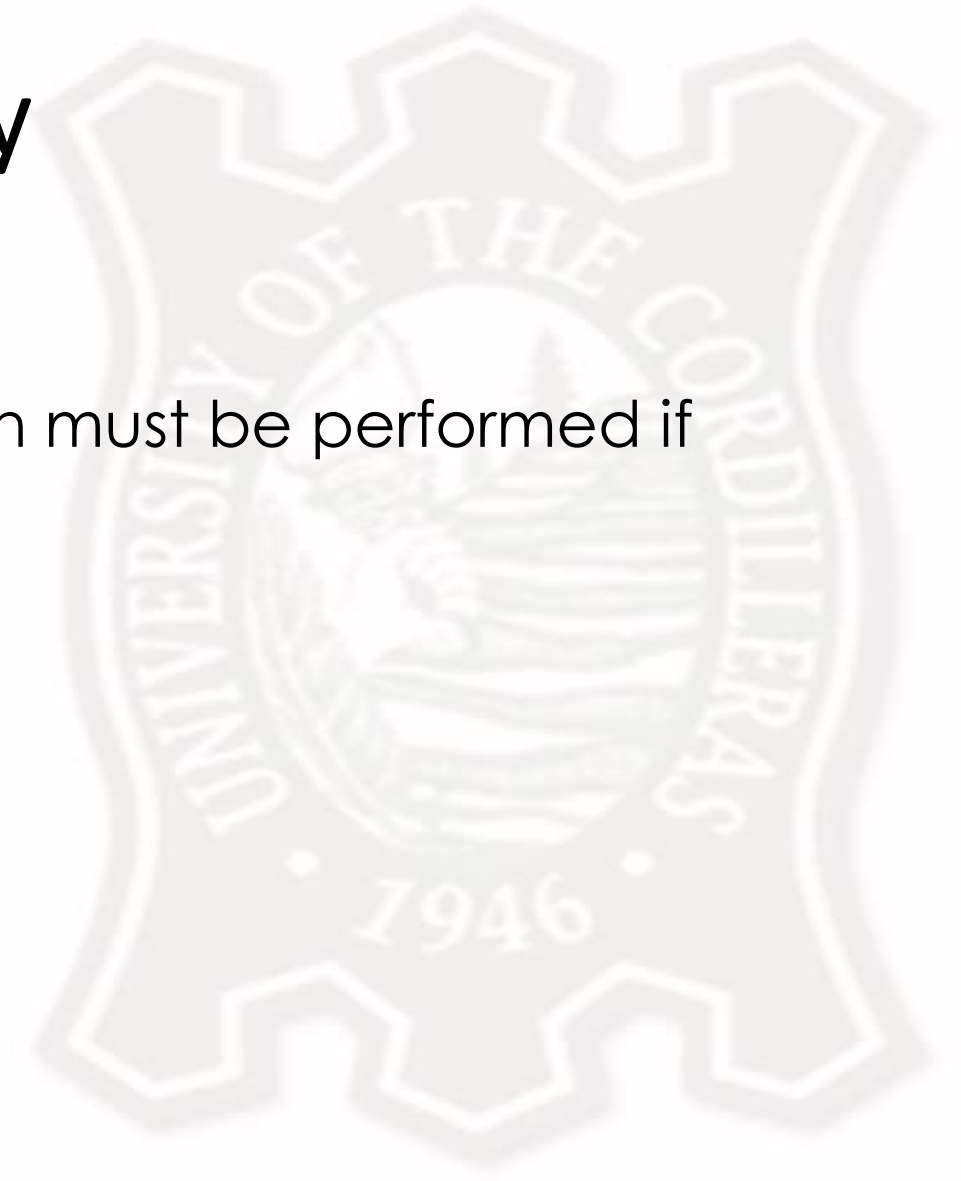
Prefix and Postfix Expressions

- Infix expressions are easily understood for humans, but not for machines
  - There are a lot of rules and regulations that cannot be easily translated into machine code
- Prefix and postfix expressions are notations that are understood better by machines
  - All rules on precedence and associativity are removed
  - All symbols can be placed on a stack
  - Increases overall efficiency of the code
  - Less understood by humans

Postfix Expressions

What is a postfix expression? | Steps – infix to postfix | Postfix expression – examples

What is a Postfix Expression?

- Notation for expressions where:
  - Operators are placed on the right side (after)
  - Operands are placed on the left side (before)
- Order of evaluation of expressions is always left to right
- Brackets cannot be used to change the order
- More commonly used for machine code
  - Translation from infix expression is much easier

Steps – Infix to Postfix

- Determine the order of operations using the rules from infix expressions
- First operands that must be evaluated are placed at the leftmost portion of the notation
- First operator that must be evaluated are placed after the operands
- Next operators and/or operands are placed at the left side of the notation
- Last operator must be at the rightmost side of the notation

Postfix Expression - Examples

- a + b = **a b +**

- c + d * 2 = **c d 2 * +**

- 3 – e + 1 = **3 e – 1 +**

- (x + y) * (z – 5) = **x y + z 5 -***

Prefix Expressions

What is a prefix expression? | Steps – infix to prefix| Prefix expression – examples

What is a Prefix Expression?

- Notation where the operators are written before the operands
- Operators act on the two nearest values on the right
  - Technically evaluated from left to right
  - The order changes depending whether or not elements to the right would be used

Steps – Infix to Prefix

- Determine the order of operations using the rules from infix expressions
- First operands that must be evaluated are placed at the rightmost portion of the notation
- First operator that must be evaluated are placed before the operands
- Next operators and/or operands are placed at the right side of the notation
- Last operator must be at the leftmost side of the notation

Prefix Expression - Examples

- a + b = **+ a b**

- c + d * 2 = **+ c * d 2**

- 3 − e + 1 = **- 3 + e 1**

- (x + y) * (z − 5) = ***+ x y − z 5**

College of
Information Technology
and **Computer Science**

CENTER OF EXCELLENCE
in Information Technology