Expressions

Last updated 8/31/21

Expressions

- Some Terminology
  - Value of an expression
    - Seems obvious
    - Subtle quirk – not necessarily the result
  - Evaluate
    - Determine the value of an expression

Expressions

- Expression: Sequence of Operators and Operands that reduce to a single value
  - Simple and Complex Expressions
  - Subject to Precedence and Associativity
  - Six categories
    - Primary
    - Postfix
    - Prefix
    - Unary
    - Binary
    - Ternary

Expressions

- Simple Expressions
  - Only 1 operator
  - a + b
- Complex Expressions
  - Multiple operators
  - 2*3/6

Expressions

- Primary Expressions
  - One operand and no operators
  - Name

    a interest_rate RATE initial1
  - Literal

    2 123.456 'a' "hello ee1910"
  - Parenthetical

    Anything in parentheses reduces to a single value

    (2 + 3 * 4) (a = b + c)

Expressions

- Postfix Expressions (only 3)
  - One operand followed by one operator
    - Operand must be a variable
  - Function Call

    Function name is an operand (named entity)

    Parenthesis are the operator

    printf(…)
  - Postfix increment/decrement

    i++ → i = i + 1

    j-- → j = j - 1

Expressions

- Postfix Expressions
  - Some expressions have a Value and a Side Effect

```
int j;
j = 5;

... j++;
```

Value of j++: 5
Side Effect: j = 6

```
Consider:
printf("%d\n", j++);
printf("%d\n", j);
→
5
6
```

  - Postfix indicates to operate after the evaluation

Expressions

- Prefix Expressions (only 2)
  - One operator followed by one operand
    - Operand must be a variable
  - Prefix increment/decrement

    ++j → j = j + 1

    --k → k= k - 1

Expressions

- Prefix Expressions
  - Some expressions have a Value and a Side Effect

```
int j;
j = 5;

…  ++j;
```

Value of ++j: 6
Side Effect: j = 6

Consider:

```
printf("%d\n", ++j);
printf("%d\n", j);
→
6
6
```

  - Prefix indicates to operate before the evaluation

Expressions

- Unary Expressions
  - One operator followed by one operand
    - Operand can be any expression

+, -
cast
sizeof(int). …

a = 5
+a → +5
-a → -5

Note: the expression is modified
not the variable, a = 5 in both cases

Expressions

- Binary Expressions
  - Operand operator operand
    - Familiar to us: +, -, *, /
    - New: % - modulus (remainder of an integer division)
    - Subject to type limitations

10 * 2 → 20	10 / 2 → 5	
true * 2 → 2	true / 2 → 0	???
'A' * 2 → 130	'A' / 2 → 32	???
15.6 * 2 → 31.2	15.6 / 2 → 7.8	
3 / 5 → 0	12 / 5 → 2	
3 % 5 → 3	12 % 5 → 2	

Modulo only operates on integers

Expressions

- Binary Expressions
  - Special binary expression - assignment
    - variable = expression
    - Has both a value - value of right side
    - And a side effect – places value into the variable on the left side
  - Simple

    a = b + c j = j * 2
  - Compound

    *=, /=, +=, -=, %=

    a *= b → a = a * b

    a += 10 → a = a + 10

    a -= b + c → a = a – (b + c) expression is evaluated first

    a /= b + c → a = a / (b + c) expression is evaluated first

Expressions

- Ternary Expressions
  - Not covered here

