Expressions

CS4700

Kenneth Sundberg

Expression

- Means of expressing computation
- Combination of values and operators that has a value

Operator Overloading

- Can users overload operators?
- Does the language overload operators?
- How does this affect the language?

Side Effects

- An observable change of global state made by a function call
- Examples:
  - output parameters
  - global variables
  - class variables
  - IO

Referential Transparency

- If an expression has no side effects it can be thought of as a reference to its value
- Functions are called pure if they have this property
- Four major causes of impurity
  - Error
  - Non-determinism
  - Context
  - Destruction

Short-Circuit Evaluation

- If the value of an expression can be and is determined without evaluating the whole expression it is called short-circuit evaluation
- Very common with boolean algebra
  - $0 * X = 0$
  - $1 + X = 1$
- Also possible with memoized pure functions

Lazy vs. Eager Evaluation

- Eager evaluation - expressions become values at earliest opportunity
- Lazy evaluation - expressions become values at the latest possible moment
  - Very common in functional languages
  - Allows expression of infinite objects - but not their evaluation

Arithmetic Expressions

- Unary
- Binary
  - Infix
  - Prefix
  - Postfix
- Ternary

Boolean Expressions

- Comparisons
  - Two way comparisons
  - Three way comparison
- Boolean algebra

Assignment

- Procedural Languages - Write to memory, always a side effect
- Functional and Logic - Create a new name binding to a constant value

Type Conversions

- narrowing
- widening
- casting
- Which operations are allowed?
- Which are implicit and which explicit?
- Are mixed mode expressions allowed?

Textbook sections covered:

- Section 07-01 (frame 2)
- Section 07-02 (frame 8)
- Section 07-03 (frame 3)
- Section 07-04 (frame 11)
- Section 07-05 (frame 9)
- Section 07-06 (frame 6)
- Section 07-07 (frame 10)