Chapter 7 – Expressions and Assignment Statements

* Introduction
  + Expressions are the fundamental means of specifying computations in a programming language
* Arithmetic Expressions
  + Automatic evaluation of arithmetic expressions similar to those found in mathematics, science, and engineering was one of the primary goals of the first high-level programming languages
  + Unary operator- operator with a single operand
  + Binary operator- has two operands
  + Ternary- three operands
  + Infix- binary operator that appears in between its operands (Most languages)
  + Prefix- binary operator that appears before the operands (Perl)
  + Precedence
    - Operator precedence rules- partially define the order in which the operators of different precedence levels are evaluated
    - Identity operator- unary addition, usually has no association operation and thus has no effect on its operand
  + Associativity- which operator is evaluated first
  + Parenthesis
  + Ruby Expressions- Ruby is pure OO language, and supports the collection of arithmetic and logic operations that are included in the C-based languages
  + Expressions in Lisp- all arithmetic and logic operations are performed by subprograms
  + Conditional Expressions- if-then-else
  + Operand evaluation order
  + Side effect- occurs when the function changes either one of its parameters or global variable
  + Referential transparency- if any two expressions in the program that have the same value can be substituted for one another anywhere in the program, without affecting the action of the program
* Overloaded Operators
  + Operator overloading- arithmetic operators that are often used for more than one purpose (ex. + being used for integer addition, floating point addition, string concatenation, etc.)
* Type Conversions
  + Narrowing conversion- converts a value to a type that cannot store even approximations of all the values of the original type
  + Widening conversion- converts a value to a type that can include at least approximations of all the values of the original type
  + Coercion in Expressions
    - Mixed-mode expressions- expressions where an operator can have operands of different types
  + Explicit Type Conversion
    - Casts- capability of a language to do explicit conversions, both widening and narrowing
  + Errors in Expressions
    - Overflow- result of an operation cannot be represented in the memory cell where it must be stored, because it is too big
    - Underflow- result of an operation cannot be represented in the memory cell where it must be stored, because it is too small
    - Exceptions- run-time errors (ex. floating-point overflow, underflow, and division by zero)
* Relational and Boolean Expressions
  + Relational Expressions
    - Relational Operator- operator that compares the values of its two operands
  + Boolean Expressions
    - Consists of Boolean variables, Boolean constants relational expressions, and Boolean operators
* Short-Circuit Evaluation- the result is determined without evaluating all of the operands and/or operators
* Assignment Statements
  + Simple Assignments- the equal sign is the assignment operator for nearly all currently used programming languages
  + Conditional Targets
    - Compound Assignment Operator- shorthand method of specifying a commonly needed form of assignment
  + Unary Assignment Operators
    - C-based languages, Perl, Javascript- two special unary arithmetic operators that are abbreviated assignments: ++ and – (increment and decrement)
* Mixed-Mode Assignment
* Summary
  + Expressions consist of constants, variables, parentheses, function calls, and operators
  + Assignment statements include target variables, assignment operators, and expressions
  + The semantics of an expression is determined in large part by the order of evaluation of operators
  + The associativity and precedence rules for operators in the expressions of a language determine the order of operator evaluation in those expressions
  + Operand evaluation order is important if functional side effects are possible
  + Type conversions can be widening or narrowing. Some narrowing conversions produce erroneous values
  + Implicit type conversions, or coercions, in expressions are common, although they eliminate the error-detection benefit of type checking, thus lowering reliability
  + Assignment statements have appeared in a wide variety of forms, including conditional targets, assigning operators, and list assignments