Chapter 2- Types, Operators, and Expressions

* Variables and constants- basic data objects manipulated in a program
* Declarations- list the variables to be used, and state what type they have and what their initial values are
* Operators- specify what is to be done to them
* Expressions- combines variables and constants to produce a new value
* Object type- determines the set of values it can have and what operations can be performed on it
* Variable Names
  + First character MUST be a letter
  + Case sensitive
  + Cannot use keywords as a variable name (*if*, *else*, *int*, etc.)
* Data Types and Sizes
  + Few basic data types in C
    - char- 1 byte, capable of holding one character
    - int- an integer
    - float- a single-precision floating point
    - double- a double-precision floating point
* Constants
  + Integer constant 1234 is an int
  + Long constant is written with a terminal L (123456789L)
  + Value of an integer can be specified in decimal, octal, or hex
    - Decimal- 31
    - Octal- add a leading 0 (3110 = 378­)
    - Hex- add a leading 0x (3110 = 0x1f16)
  + Char constant- written with single pair of ‘’ (ex. char example = ‘x’;)
    - ‘x’ is NOT the same as “x” (char =/= String)
* Declarations
  + All variables must be declared before use
* Arithmetic Operators
  + +, -, \*, /, % (modulus)
* Relational and Logical Operators
  + Highest precedence: >, >=, <, <= (left side or right side can have more value over the other)
  + Equal precedence: ==, != (Both sides are either equal or not equal)
  + Least precedence: && and :: (always evaluated left to right)
* Type Conversions
* Increment and Decrement Operators
  + If n = 5, and:
    - x = n++; // sets x to 5
      * n is incremented after its value is used.
    - x = ++n; // sets x to 6
      * n is incremented before its value is used.
* Bitwise Operators
  + & bitwise AND
  + : bitwise inclusive OR
  + ^ bitwise exclusive OR
  + << left shift
  + >> right shift
  + ~ one’s compliment
* Assignment Operators and Expressions
  + i = i + 2;
    - Shorthand: i += 2;
* Conditional Expressions
  + If (a > b) … else …
* Precedence and Order of Evaluation