**Chapter 10: General Issues in Using Variables**

**Data Literacy**

* The first step in creating effective data is knowing which kind to create

**Making Variable Declarations Easy**

* Don’t use implicit declarations
* Use naming conventions
* Cross reference variable names

**Scope**

* How famous a variable is?

Localize References to Variables

* Good idea to localize references to variables by keeping them close together in code
* Short number of lines in between uses
* Improves readability

Keep Variables “Live” for as Short a Time as Possible

* Number of statements over which a variable is live
* Want to keep low
* Gives you a more accurate picture of code as you can concentrate on smaller section
* Good for splitting large routines into smaller ones as well

General Guidelines for Minimizing Scope

* Initialize variables used in a loop immediately before the loop, instead of at beginning of routine
* Don’t assign value to variable until just before its used
* Group related statements
* Break groups of related statements into separate routines
* Begin with most restricted visibility
  + Expand only if necessary
  + Part of keeping variable as local as possible

**Persistence**

* Lifespan of a piece of data
* Lifespans
  + For life of particular block of code or routine
  + For life of program
  + Forever (database)
* Main problem is when you assume variable has longer persistence than it really does
* Steps to solve:
  + Use debugger to check for reasonable values
  + Set variable to “unreasonable values” when done with them
  + Write code that assumes data isn’t persistent
  + Develop habits of declaring and initializing data right before its used
    - If you see data used without a nearby init, be suspicious

**Binding Time**

* Time at which the variable and its value are bound together
* Ideally want to make binding time as late as possible
  + Most flexibility
* Times when variable van be bound:
  + Coding time (magic numbers)
  + Compile time (named constant)
  + Load time (read from external source)
  + Object instantiation time (read value each time window created)
  + Just in time (value each time window is drawn)

**Relationship Between Data Types and Control Structures**

* Patterns in code match patterns in data
* Three types of data and corresponding control structures:
  + Sequential data translates to sequential statements in a program
    - Clusters of data used together in a certain order
  + Selective data translates to if and case statements
    - One of several pieces of data used at any particular time
  + Iterative data - for, repeat, while looping structures
    - Same data type repeated several times

Using Each Variable for Exactly One Purpose

* Sometimes tempting to use one variable in two places for two diff activities
  + “temp”, “x”
* Avoid variables with hidden meaning
* Make sure all declared variables are used