Use Case Analysis and Design Quizzes

# Design

Coupling – How inter-related are the modules in a system?

* Communication
* Design goal: Minimize coupling to reduce need for communication among modules
* Coupling cannot always be avoided
* Design goal: Where coupling is unavoidable:
  + Make it a desirable form
  + Make needed communication precise
* Undesirable Forms:
  + Non-modular, make it difficult to reason about software
  + Global (or Common) coupling
    - Example: Using global variables to share information among modules
  + Content coupling
    - Example: Implementation inheritance, whereby changes to the content of a class implementation affects that of another
  + Control coupling:
    - External flag controls flow in a module
* Desirable Forms
  + Parametric coupling
    - Example: Use of parameters to communicate information among modules
  + Coupling strictly through interfaces
    - Reuse of components strictly based on interface specifications
    - Specification inheritance, whereby existing specification is extended

Cohesion – How single-minded is each module.

* Module being coherent – making sense
* Design goal: Maximize cohesion in each module
* Design goal: Make the cohesion be a “desirable form”
* Undesirable form:
  + Grouping of “unrelated elements”
  + Coincidental cohesion
    - Example: Elements of a module come together by accident
  + Logical cohesion:
    - All outputs grouped in a module
  + Temporal cohesion:
    - Elements that happen in close proximity in time are grouped together
    - Example: “Start up” module
* Desirable Forms:
  + Functional cohesion
    - Elements in module perform related functionality
    - Example: Interface specification and implementations that capture a well-designed “abstract data type” (eg stacks, queues, lists, or maps)