

## Information Hiding – D. L. Parnas

- Do NOT give a module more information than is absolutely necessary
- NEVER pass the entire record when ONLY certain fields are required
- Do NOT pass Flow of control Flags

## Information Hiding

Procedure A

    Call B(X)

Procedure C

    Call B(Y)

Procedure B(Z)

    If Z           Then.....

                  Else .....

## Structured Programming

- A program which only uses structured constructs

## Structured Constructs

- Sequence
- Assignment
- Proc, Function call
- While – do, Repeat – Until
- If – then, If – Then – Else
- Case

## Unstructured Constructs

- GOTO
- FOR
  - Why?
    - For I = J to K
    - What happens if J > K?

## Designing

- “Minimizing connections between modules also minimizes the paths along which errors can propagate into other parts of the system thus eliminating disastrous side effects.”
- Global Variables can result in an enormous number of connections between modules.”

## Global Variables

- N modules
- M Global variables
- Results
  - $M ( N ( N-1) )$  connections

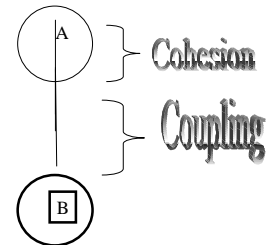
## Designing

- Complexity of a system is affected not only by the number of connections but also by the DEGREE of the connection”
- COUPLING – measure of the strength of associated connections
- “A well structured system, one in which communication is through well defined interfaces is likely to be more growable and require less effort to maintain than one using global variables”

## Reduce Coupling

1. Minimize relationships among modules
2. Maximize relationships among elements within the module
3. Strive for high COHESION

## Cohesion



## Levels of cohesion

1. Coincidental – no meaningful relationship among elements
  1. Modularizing existing code
  2. Save duplication
2. Logical – some logical relationship similar to Functional
3. Temporal – logical except related in time.
4. Communicational – related by a reference to the same data (read and print file)
5. Sequential – transformation of data
6. Functional – all elements are related to the performance of a single function.

## Why use a Design Methodology

- Create “better” software
- Methodologies claim:
  - 1.9 % increase in productivity
  - 39% savings in programming cost
  - 80% reduction in bugs
- *How are these studies done?*
- *Are these claims true?*

## General Design Strategies

1. Yourdon’s structured Design
2. Jackson’s Data Structure
3. Warrier – Orr diagrams