2 Levels of Design

1. High level: architecture design

* structure of system: components, subsystems, modules, classes
  + software quality measure based on this
* EX: OOA/OOD/OOP (Object Oriented)
  + similarity: all share object concepts
  + difference: different phases, can use/not use any, not in sequence

Analysis: JSD: id entity/object, services

from practice => spec by object

Design: from specification => design around objects

Programming: Java, C#

* moduling decision: divide & conquer
  + based on functionality/interests (not race, gender)
  + make each module/class as small as possible
    - if large, then probably squish too many things together
  + Cohesion: relation of code chunks within a module/class, so close that they can’t be distinguished
  + Coupling: relation between 2 modules, loose so that modules are very distinct
* EX: CPU
  + ALU, Memory/Storage Unit, Control Unit (processor, memory, I/O)

2. Low level: detail design

* data structure/algorithm

4 Levels of Cohesion (Worst to Best)

1. Trash/Garbage bin: Coincidence
   * impossible to reuse
   * occur by chance
   * EX:
     + eat breakfast
     + submit assignment
     + go to sleep
2. Logical level
   * share some physical feature
   * not possible to reuse therefore unacceptable
   * EX:
     + take Engl101
     + take CSc221
     + take Phil300
   * Low Kolmogorov Complexity: take 3 different classes
3. Temporal: Association
   * actions mostly occur in sequence (time)
   * typically happens like this
   * EX:
     + Quarterback throw ball
     + Receiver take ball and run
     + Touchdown
   * not procedural bc even though typical, isn’t always what happens
   * EX:
     + choose book
     + pay cashier
     + take book out
   * EX:
     + cheat in final exam (not get caught)
     + get F from class
   * not trash bin bc if caught cheating then get F, still linked
4. Procedural: Cause and Effect
   * prior actions are pre-condition to later actions
   * much closer
   * EX:
     + take 221
     + take 322 (pre-reqs)

//In Final, explain justification

Double-blindness testing

Google vs Bing

Examples

1. Situation:
   * One semester, 3 A+: > 6’2”
   * very tall boys => get A+ in computer vision
   * Cohesion: trash bin
     + no replication
     + very low probability
2. Situation:
   * 3 A+: students w GPA > 3.8
   * Cohesion: temporal
     + not procedural bc not supposed to know GPA
3. Situation
   * read textbook
   * print out assignments
   * drop class
   * see advisor
   * Cohesion: logical
     + cannot reuse
     + share something in common, student activities
4. Situation
   * print out resume
   * practice algorithm questions
   * ask friend to test your knowledge in coding
   * Cohesion: logical
     + slim connection: prepare for interview

Levels of Communication

5: All actions share the same data/file

* Before OO:
  + module: deal with one data structure only
* data file is more stable than actions
* communication:
  + input data entries to a DB
  + query DB based on certain keyword
  + save DB back

Best lvl

6a: one function for one module (non OO)

* short module

6b: well-defined class definition (OO only)

* 3NF: no data redundancy

7 Levels of Coupling (Worst to Best)

1. Content: one module can directly access the private contents of the other friend
   * too close, very bad
   * generally unacceptable
   * for convenience of cin, cout
2. Common: share the same global variable, pretty close,
   * all functions linked, hard to localize
   * EX:

Module A:

...

time = 100;

Module B:

cout << time;

time += 10;

* + C/C++/C#: cinch
  + Java: can be done, not easy, purposely difficult to not use, must justify
  + avoid using

1. Control: returned value of one module determines different logic flow in another module
   * EX

Module A:

...

return value

Module B:

...

val = (mod) A(...)

if val > 1:

...

print(‘fine’)

else:

...

print(‘no good’)

* + from B, treat A as one black box
  + only know return value, nothing else; pretty loose
  + acceptable, no need to justify

1. Stamp: have communication, not all info are useful
   * EX:
     + A sends a student obj to B
     + in B, just print out name & GPA
2. Data: send only necessary info
   * EX:
     + A sends a student’s name & GPA to B
     + in B, prints out name & GPA
3. No Direct Relation
4. Stop

On Final

Given: Modules A, B, C, D

Explain: AB, AC, AD, BC, BD, CD

Optional: Draw Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| A | X |  |  |  |
| B | X | X |  |  |
| C | X | X | X |  |
| D | X | X | X | X |

symmetric, choose one half (lower/upper)

assign worst coupling