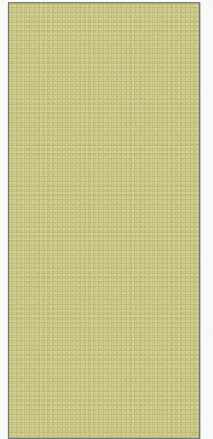


# INTRODUCTION TO PROBLEM SOLVING

HOW DO WE SOLVE A PROBLEM?



# THE PROGRAM DEVELOPMENT PROCESS

# STEPS IN THE PROGRAM DEVELOPMENT PROCESS

1. Structured Walkthrough
2. Stepwise Refinement
3. Modular Design
4. Bottom-Up Coding
5. Testing
6. Documentation

# STRUCTURED WALKTHROUGH (1)

- AKA Requirement Analysis
- Before programming sit down with client and discuss requirements
- Requirements often imprecise in the beginning
- May require **several discussions** to come up with clear requirements

# STRUCTURED WALKTHROUGH (2)

- Enumerate I/O specifications, processes, and constraints
- In the end, problem specifications must be
  - very **precise**, and
  - understandable by the programmer in **programming terms**
- VERY IMPORTANT!

# STEPWISE REFINEMENT (1)

- AKA Top-Down Design or Divide and Conquer
- Break the problem into smaller parts
- Break these smaller parts to even smaller parts
- Repeat until each part can be easily done



# STEPWISE REFINEMENT (2)

- Tips
  - Divide jobs into three parts
    - Input
    - Process
    - Output
  - Draw a tree and refine each job level by level (breadth-first)
  - Use pseudo code to describe each job
  - Decision on data structure is delayed as much as possible

# STEPWISE REFINEMENT (3)

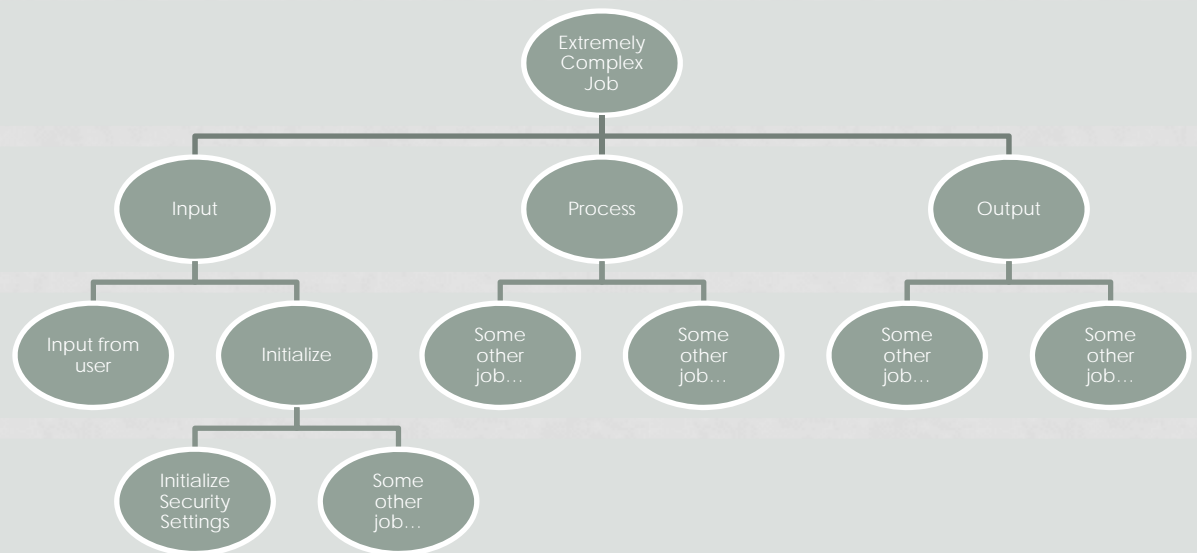
Level 1

Level 2

Level 3

Level 4

...





# MODULAR DESIGN

- Define input and output specifications of each module
- Each ellipse in the tree is a module
- Each module is a self-contained block
  - Receives input from immediate ancestor
  - Outputs to immediate ancestor
  - Should only call modules that its immediate children
- Algorithm for each module: flowcharts and pseudocodes
- Reuse codes and data structures

# BOTTOM-UP CODING

- Code the simplest function first
- Simplest function can be tested independently
- In a team, the project manager can distribute labor among the team members by giving them functions to program independently

# TESTING USING STUBS

- Field testing before deployment
- Stubs
  - Use (human) testers to emulate functions of unfinished modules
  - Tester acts on inputs and enters the correct output for the finished modules of the program
- Programmers can test their work even when other programmers are not yet finished
- Programmer of unfinished module will finish it

# WHITE BOX AND BLACK BOX TESTING

- Done for each module and for the whole program
- White Box
  - Input values and observe output values
  - Tester must know the result for a given set of input values
  - Practical for small systems only
- Black Box
  - Given a set of input values, is the result reasonable?
  - The result should not be known beforehand

# STRUCTURED PROGRAMMING DOCUMENTS

- Generated along with the program
- Describes the whole program and each module in the program
- When a requirement changes, check the document
- Only the modules that need to be modified and the subtree under those modules need to be modified
- Useful since a programmer forgets his code after some time (2 months)



# QUIZ # 1 (1/4)

1. What is another term for Stepwise Refinement?
2. Fill in the blank. In bottom-up coding, the \_\_\_\_\_ module is programmed first.
3. True or False. Each module should be programmed independently.
4. Differentiate White Box and Black Box Testing.