

CMSC 21

Fundamentals of Programming

2nd Semester 2011-2012

INTRODUCTION TO PROBLEM SOLVING

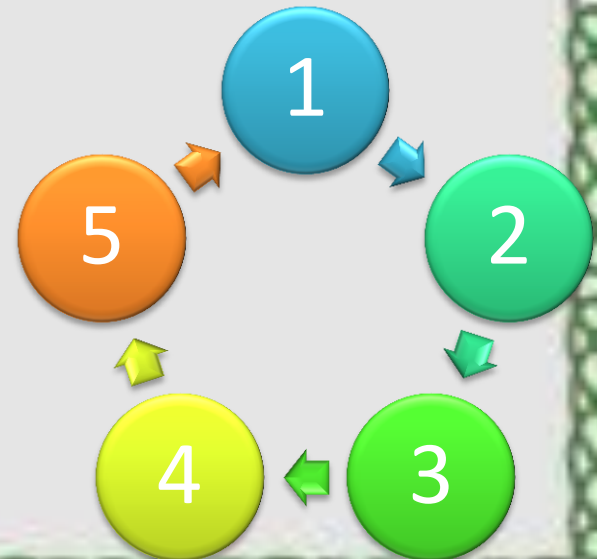
“Resist the temptation to code”

- Old programming proverb

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



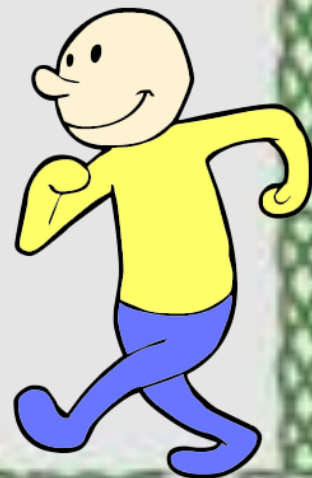
Structured Walkthrough

- Also known as REQUIREMENT ANALYSIS
- Understanding the problem through its **specifications**
- Involves **asking questions** to the client
- May require **several discussions** to come up with clear requirements



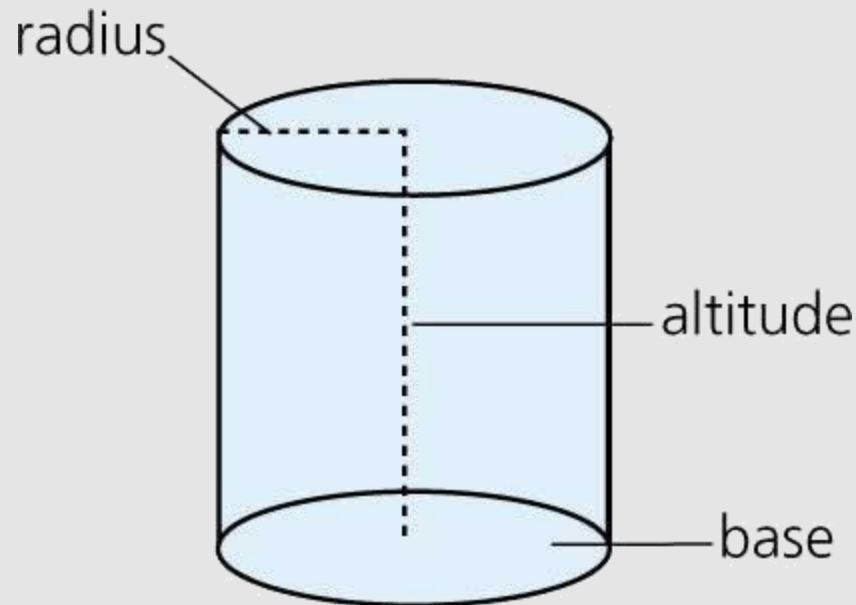
Structured Walkthrough

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



Structured Walkthrough

Example: Get the surface area of a cylinder



Academy Artworks

Structured Walkthrough

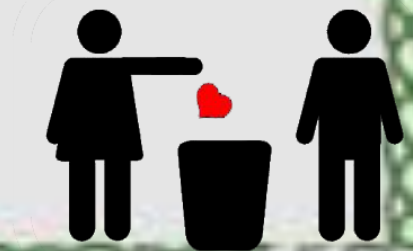
- Suppose we do not know the formula for computing the area of the cylinder
- All we know are the following:
 - We can compute for the area of a circle
 - We can compute for the area of a rectangle
 - The radius and the height

Stepwise Refinement

- Also known as Top-Down Design or Divide and Conquer
- Break the problem into smaller parts until each part can easily be done

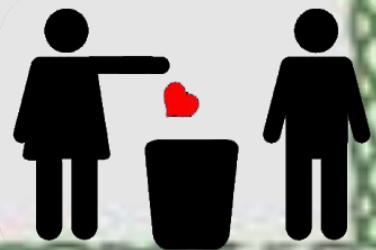
Hoare's Law Of Large Problems

"Inside Every Large Problem Is A Small Problem Struggling To Get Out."

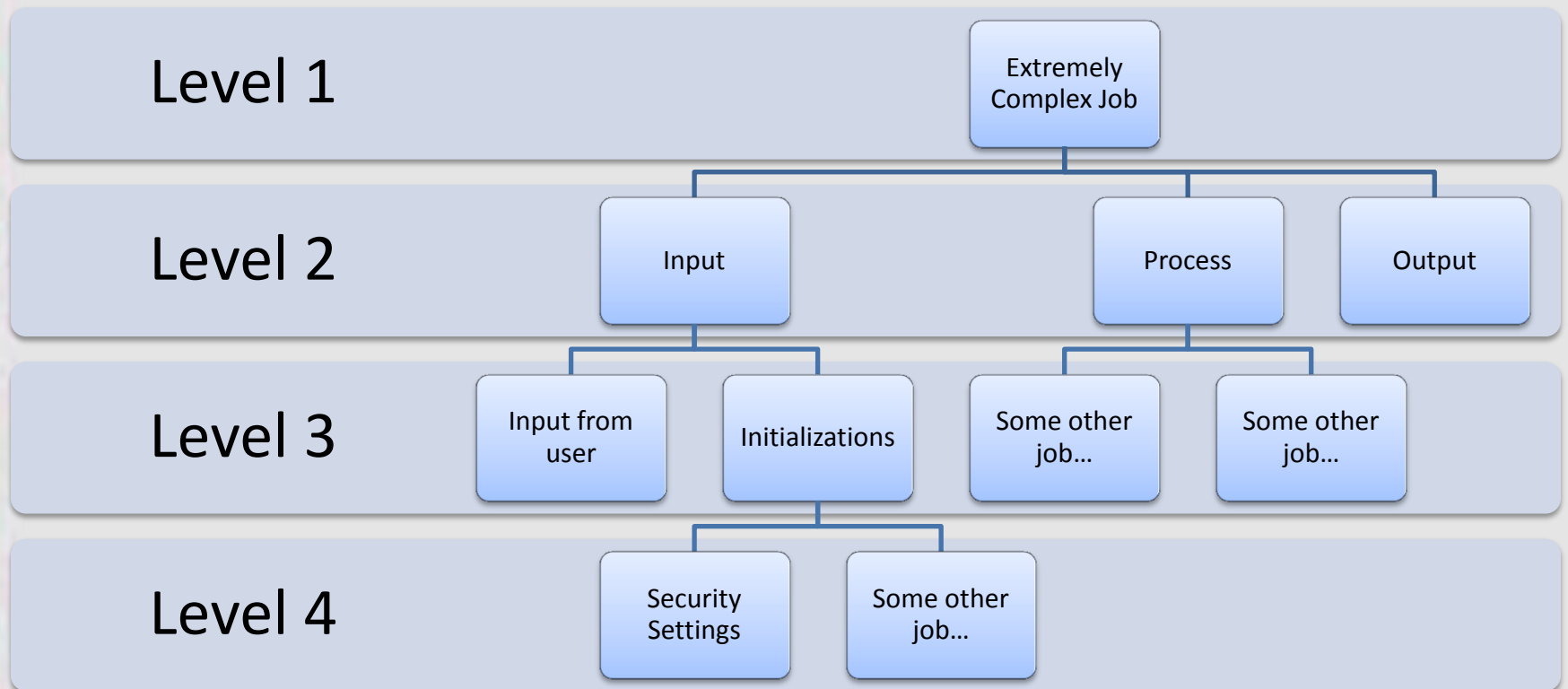


Stepwise Refinement

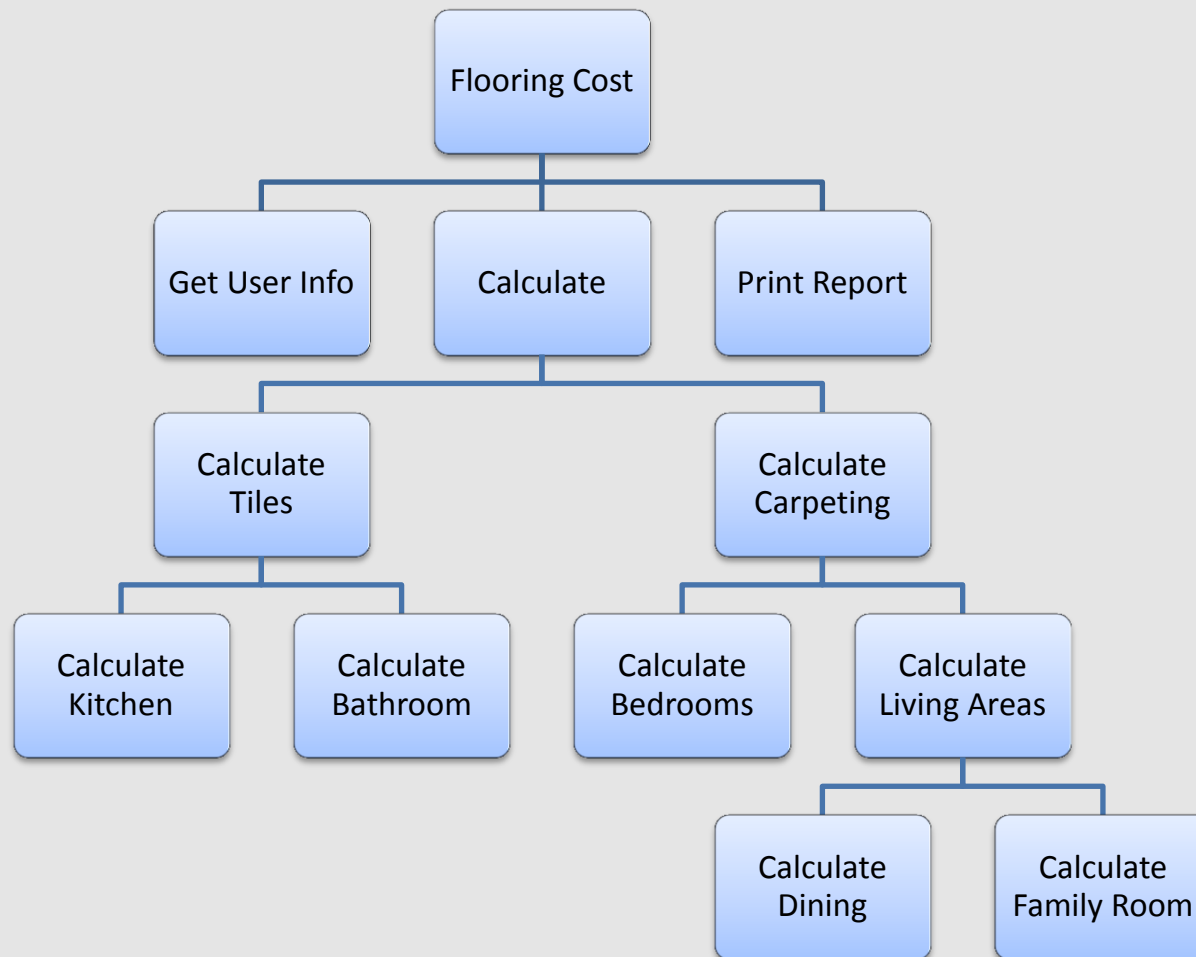
- Divide the jobs into three parts
 - Input
 - Process
 - Output
- Draw a tree and refine each job level by level (breadth-first)
- Use pseudo code to describe each job
- Decision on data structure is delayed as much as possible



Stepwise Refinement



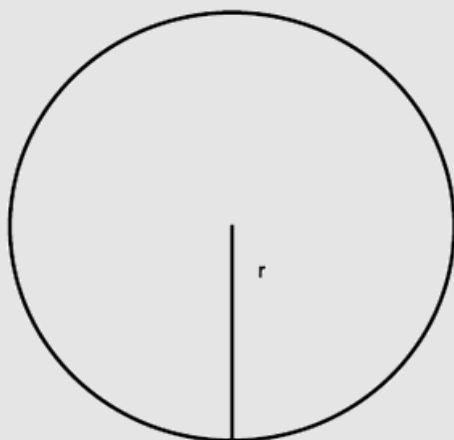
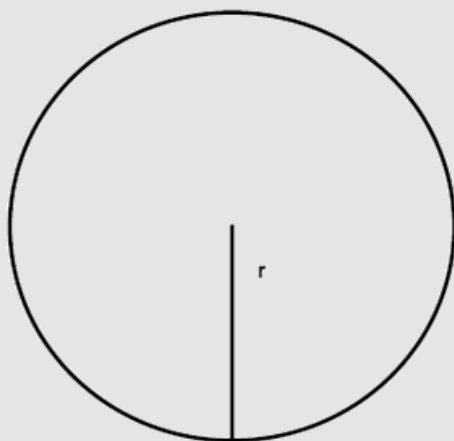
Stepwise Refinement



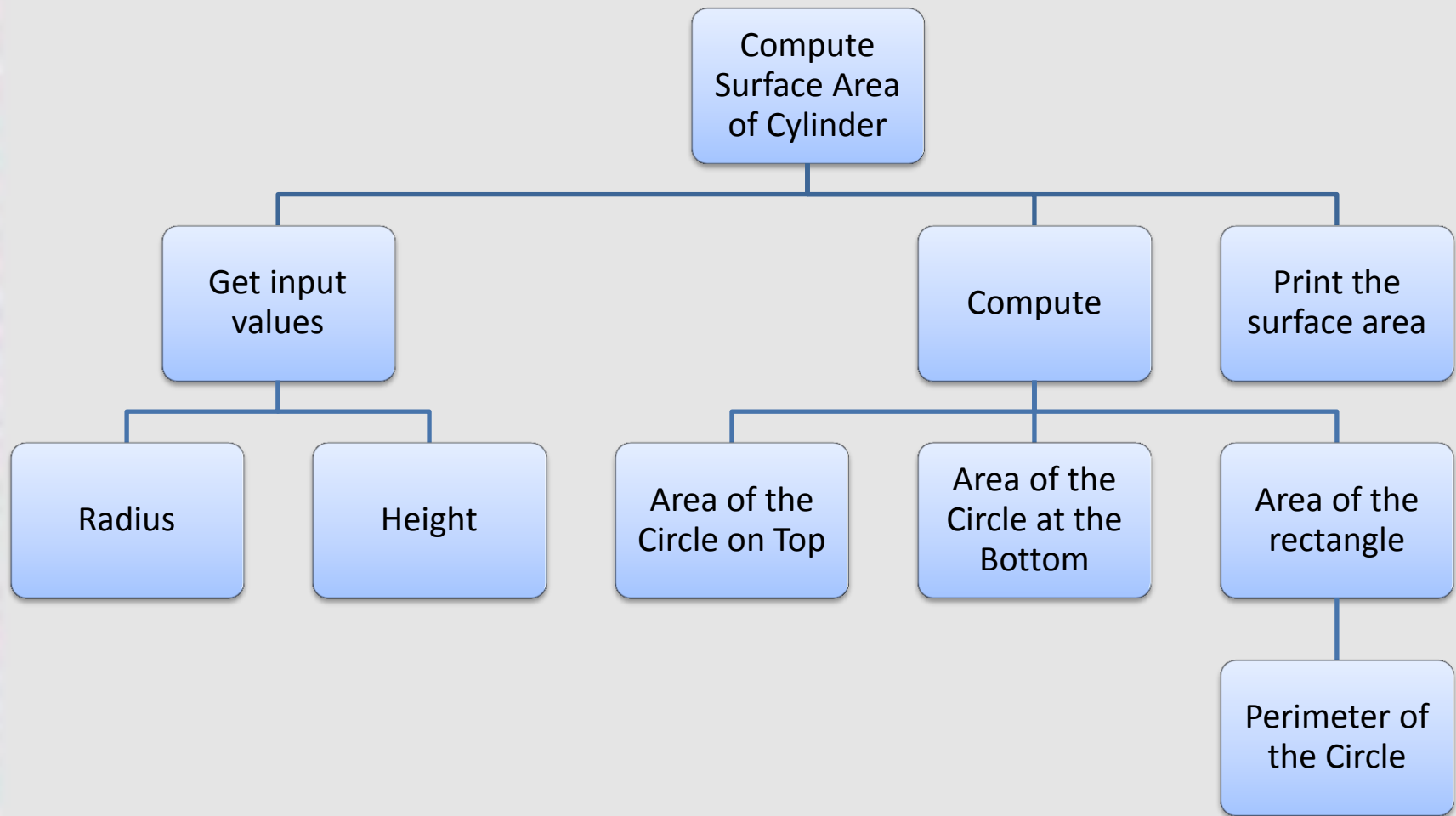
Stepwise Refinement

- We divide the problem into three parts:
 - Input: The radius and the height
 - Process: Compute for the area of two circles and the area of a rectangle.
 - Output: The surface area of the cylinder

Stepwise Refinement



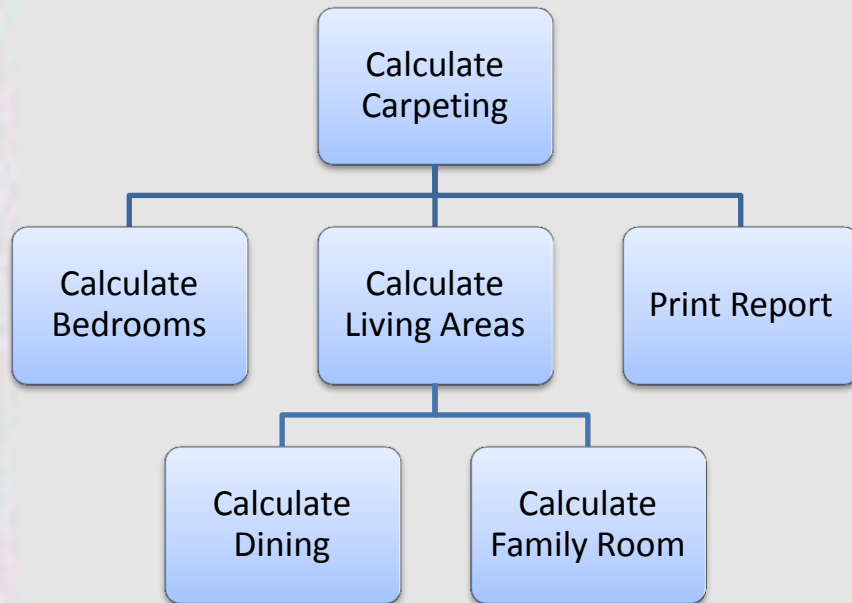
Stepwise Refinement



Modular Design

- Define the input and output specifications of each module
- Design an algorithm for each module using flowcharts and pseudocodes
- Each module is a self-contained block
 - Receives from the immediate ancestor
 - Outputs to immediate ancestor
 - Should only call modules that are its immediate children
- Codes and data structures ought to be reusable

Modular Design



- Calculate Living areas is the immediate ancestor of Calculate dining ad Calculate Family Room
- Flooring cost for Living Areas = Flooring Cost for Dining + Flooring Cost for Family Room

Modular Design

- Each node in the hierarchy is a module
 - getRadius – accepts radius as an input
 - getHeight – accepts height as an input
 - computeAreaOfCircle – calculates the area of a circle given the radius
 - computeAreaOfRectangle – calculates the area of the rectangle given the height and the width
 - computePerimeterOfCircle – calculates the perimeter of the circle

Modular Design

- Each node in the hierarchy is a module
 computeSurfaceAreaOfCylinder – calculates the
 surface area given the area of circles and the area of
 the rectangle
 printSurfaceArea – outputs the result

Bottom-up Coding

- Code the simplest function first
- Simplest functions can be tested independently
- Modules can also be distributed among team members



Program Testing

- Testing Using Stubs
- White-box testing
- Black-box Testing



Testing Using Stubs

- Fields testing before deployment
- Stubs
 - Use (human) tester 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



White-Box Testing

- Assumption: the tester knows everything about the program
- Given a set of inputs, the tester must know whether what output to expect
- Practical for small systems only



Black-Box Testing

- Assumption: the tester knows nothing about the program
- The result for a set of input should not be known beforehand
- Does the program give a reasonable set of output given a set of test input?



Documentation

- Generated along with the program
- Describes the whole program and each module in the program
- Helps in identifying which modules are needed to be modified when the requirements changed
- Useful since a programmer forgets his code after some time



QUIZ (1/4)

1. Differentiate White-box testing from Black-box testing
2. It is a program design tool that is a visual representation of the logic in a function within the program

