



CMSC 21 Fundamentals of Programming

Structured Programming



Program Development Process via Structured Programming

- Structured Walkthrough (Requirement Analysis)
- Stepwise Refinement (Top-down Design)
- Modular Design
- Bottom-up Coding
- Testing (using stubs, black-box and white-box tests)
- Documentation



Structured Walkthrough or Requirement Analysis

- I/O specification of the program
- Enumerate processes and constraints
- In software engineering
 - Client's requirement is often imprecise
 - Several rounds of discussion with the client
 - Must generate specific requirements for the programmer



Stepwise Refinement or Top-down Design

Complex problems can be solved using top-down design , also known as stepwise refinement, where

- We break the problem into parts
- Then break the parts further into smaller parts
- Until each of the parts can be easily done



Modular Design

- Define the input and output specifications of each module
- Design the algorithm for each module e.g. flowcharts and pseudocodes
- Apply code reuse for each module and control structures within each module



Bottom-up Coding

- Implement of the identified modules/functions starting from the simplest
- You can test the simple function independently of the rest of the program. This gives you achievement and satisfaction, sustaining you through the entire development process.
- You can also divide and assign tasks at this point where each member of the dev team can be assigned to work on specific modules.



Testing

- Stubs - for unfinished modules, you can substitute it with “stubs” which can be in the form of hard-coded or manually generated input
- White-box Testing – input something into the system, check if it follows the expected path and then check if it generates the expected result. Only practical for relatively smaller systems.
- Black-box Testing – input something into the system and then check if the system generates the expected result. It doesn't check the path the input had gone through. For larger systems, one is forced to use black box testing.



Documentation

- Programmer usually forgets their code in 2 months; the structured document written along with the program will help a lot in guiding the programmer to understand the codes.
- When there are changes in the client's requirement, you don't need to re-write the entire program, the documentation can help you identify find which modules are only needed to be modified.