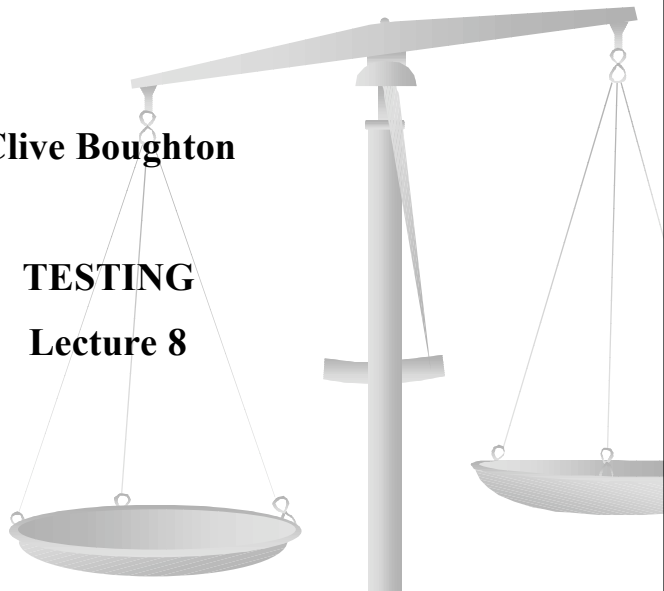


Quality Management

Clive Boughton

TESTING

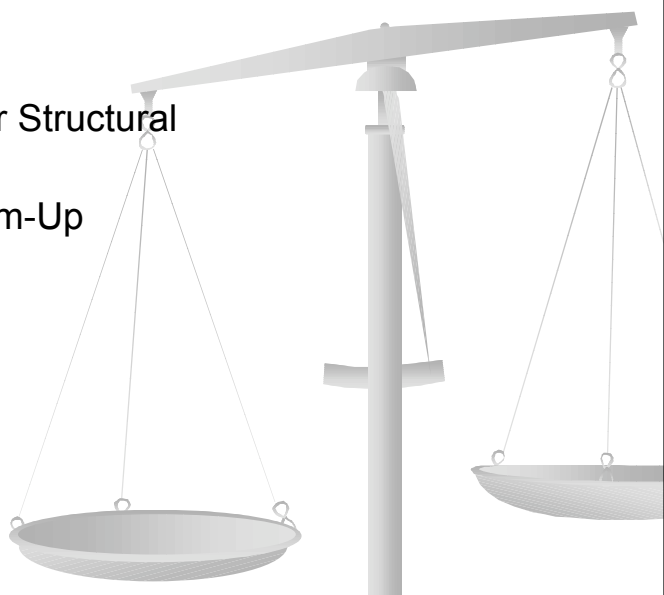
Lecture 8



Testing

Types of Test

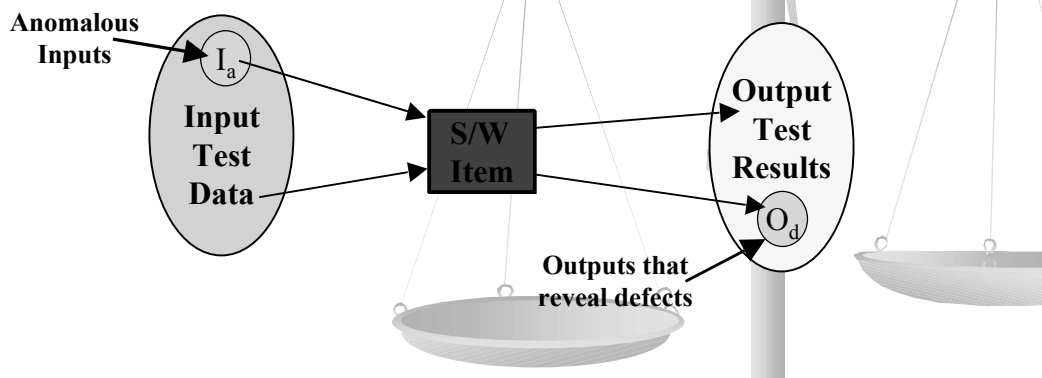
- Black Box
- White/Glass Box or Structural
- Path
- Top-Down & Bottom-Up
- Interface
- Stress
- *Object-Oriented*



Testing

Black Box Testing

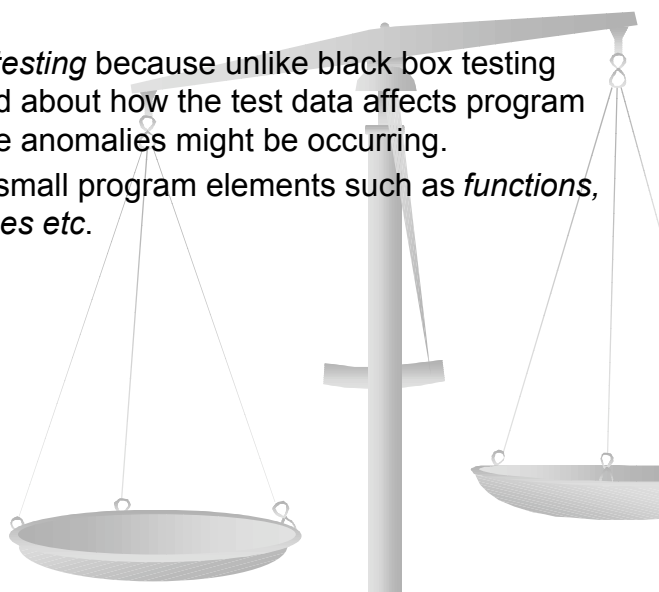
- Also called *functional testing* because the tester is only concerned about whether the test produces the right results and not how the software is implemented.



Testing

White Box Testing

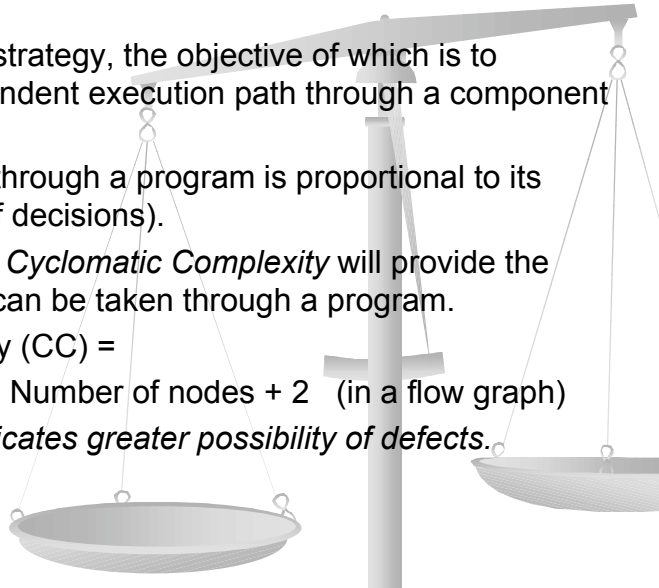
- Also called *structural testing* because unlike black box testing the tester is concerned about how the test data affects program flow and exactly where anomalies might be occurring.
- Usually only used for small program elements such as *functions, procedures and routines etc.*



Testing

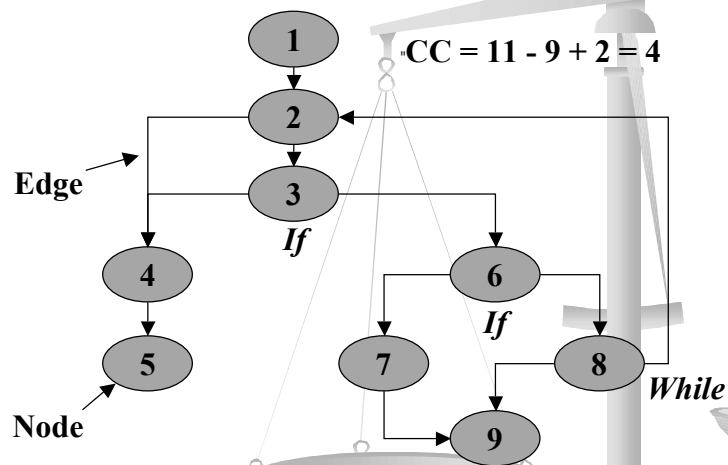
Path Testing

- Is a *structural testing* strategy, the objective of which is to exercise every independent execution path through a component or program.
- The number of paths through a program is proportional to its complexity (number of decisions).
- Calculating McCabe's *Cyclomatic Complexity* will provide the number of paths that can be taken through a program.
- Cyclomatic Complexity (CC) =
 - Number of edges - Number of nodes + 2 (in a flow graph)
- *A higher CC often indicates greater possibility of defects.*



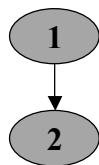
Testing

Cyclomatic Complexity (Example)

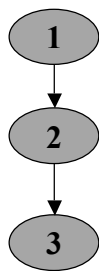


Testing

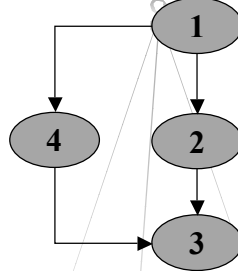
Cyclomatic Complexity (Exercise)



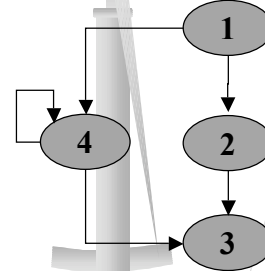
CC = ?



CC = ?



CC = ?



CC = ?

Testing

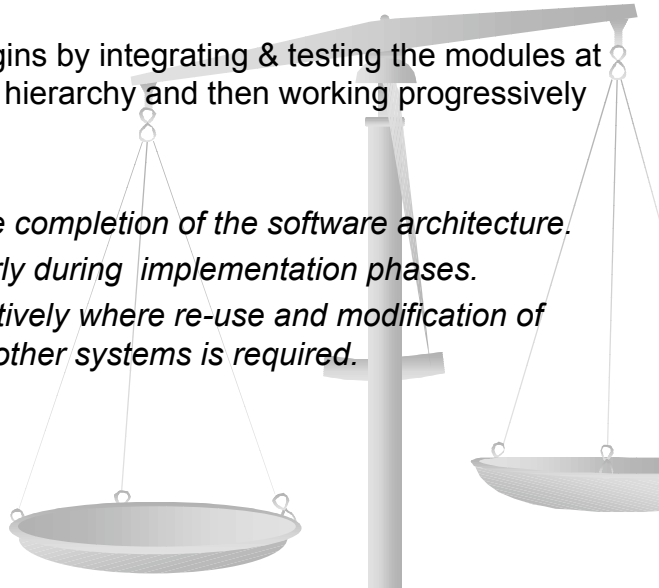
Top-Down Testing

- Are different approaches to system software integration.
- Top-Down testing integrates & tests the high level components before their design & implementation has been completed.
- Top-Down testing is an integral part of a development process where work is started on high level components & the low-level components are mere stubs.
- Top-Down testing is -
 - *Likely to uncover defects in the software architecture.*
 - *Helpful in providing progressive system demonstration.*
 - *Difficult because the stubs need to simulate their final implementations.*

Testing

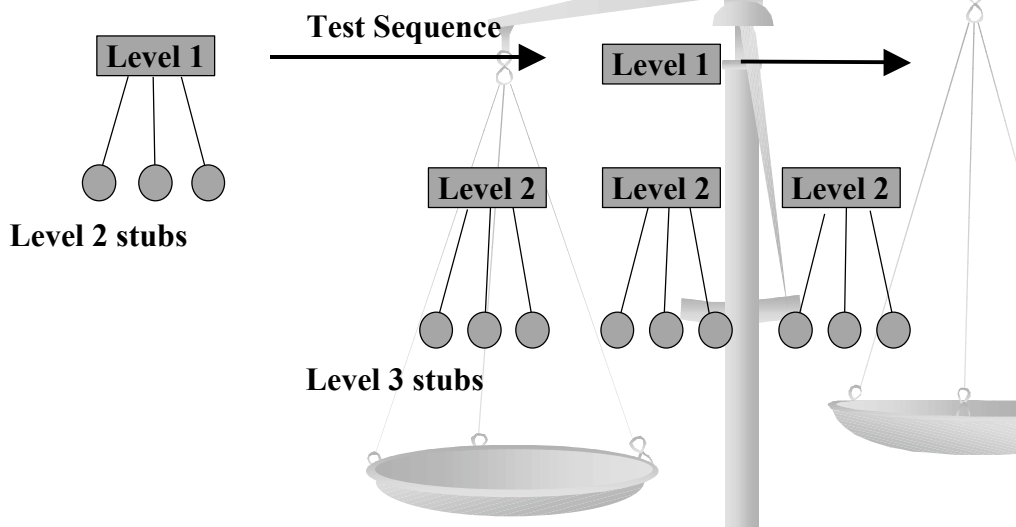
Bottom-Up Testing

- Bottom-Up testing begins by integrating & testing the modules at the lower levels in the hierarchy and then working progressively up the hierarchy.
- Bottom-Up
 - *Doesn't require the completion of the software architecture.*
 - *Can be started early during implementation phases.*
 - *Can be used effectively where re-use and modification of components from other systems is required.*



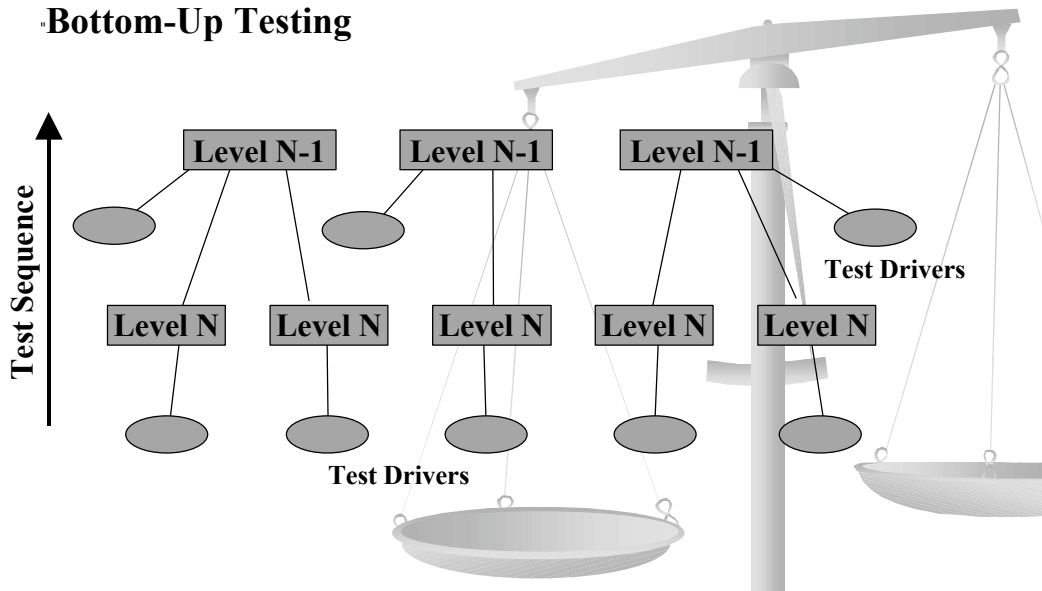
Testing

Top-Down Testing



Testing

Bottom-Up Testing



Testing

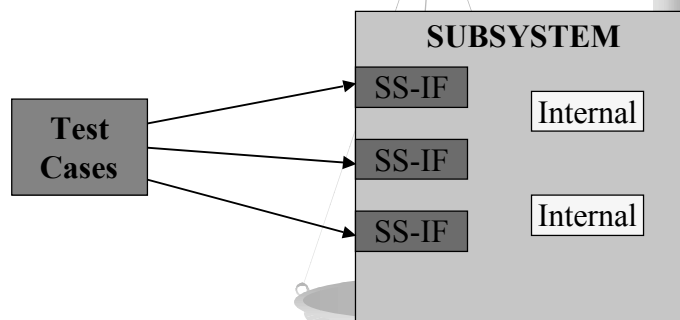
Interface Testing

- Is applied when modules or subsystems are integrated to create larger (sub)systems.
- Intent is to detect faults that might have been introduced due to interface errors or invalid assumptions about interfaces.
- Is particularly important for O-O development when objects and classes are re-used.
 - *O-O I/F errors cannot be detected by testing the individual objects because such errors are a result of the interaction between objects rather than the isolated behaviour of a single object.*
- *Some different types of interface include:*
 - *Parameter, Shared memory, Procedural and Message passing.*

Testing

Interface Testing

- Is often difficult because faults sometimes manifest themselves only under *particular/peculiar conditions*.
- Also difficult because *incorrect assumptions* may be made about valid responses that are, nonetheless, incorrect.



Testing

Stress Testing

- Is oriented toward the *determination of performance and reliability* of the integrated system.
- Such tests are usually done in a series where, with each test the *load on the system is steadily increased* until system performance is unacceptable.
- This type of test has two functions:-
 - Determination of *failure behaviour* such as graceful degradation or unrecoverable crash.
 - The *detection of defects that would not otherwise arise* or be evident under 'normal' load.