# Software Testing

# Performance Testing

- Performance tests must be designed to ensure that the system can process its intended load
- This usually involves planning a series of tests where the load is steadily increased until the system performance becomes unacceptable
- This type of testing has two functions
  - It tests the failure behavior of the system
  - It stresses the system and may cause defects to come to light that would not normally be discovered

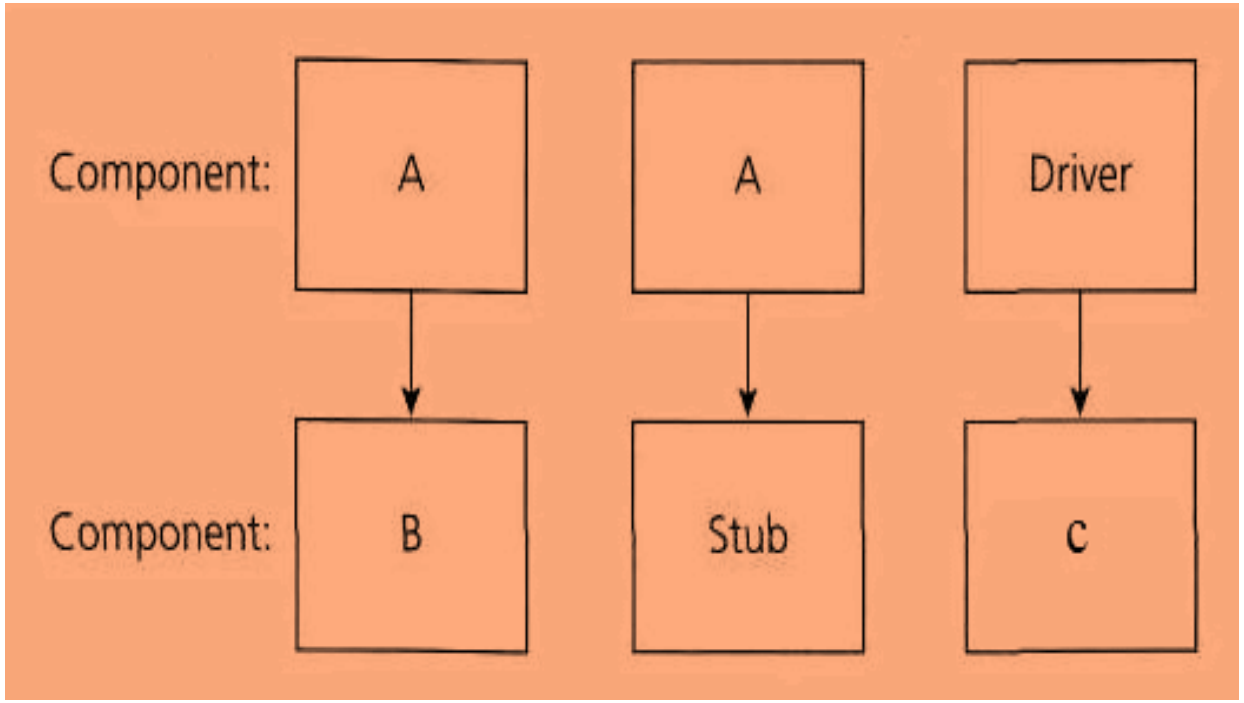
# Component Testing

- Component testing also called as unit testing is the process of testing individual components of a system
- This is a defect testing process so it goal is to expose faults in these components
- The different types of components that are tested are
  - Individual functions or methods within an object
  - Object classes that have several attributes and methods
  - Composite components made up of several different objects or functions

1. The main focus id to uncover the errors in design and implementation.
2. It may be done in isolation from rest of the system depending on the development life cycle model chosen for that particular application.
3. In such case the missing software is replaced by Stubs and Drivers and simulate the interface between the software components in a simple manner.

# Component Testing

- A stub is called from the software component to be tested.
- As shown in the diagram below 'Stub' is called by component A', and a driver calls the component to be tested.
- As shown in the diagram below 'component B' is called by the 'Driver’.



# Interface Testing

- Testing composite functions or objects (components made up of several interacting objects) is primarily concerned with testing that component
- Interface testing is particularly important for object oriented and component based development
- Interface errors in the composite component cannot be detected by testing the individual objects or components
- Errors in composite components may arise because of interaction between its parts

- The different types of interfaces  are
  - Parameter interfaces
  - Shared memory interfaces
  - Procedural interfaces
  - Message passing interfaces

- The errors are
  - Interface misuse
  - Interface misunderstanding
  - Timing errors

# Structural Testing

- It is an approach to test case design where the tests are derived from knowledge of software structure and implementation
- This approach is also called as **white box testing or glass box testing or clear box testing**
- The following figure gives the description of structural testing
- Path testing is a structural testing strategy whose objective is to exercise every independent execution path through a component or program
- The objective of path testing is to ensure that each independent path through the program is executed at least once
- The number of paths through a program is usually proportional to its size and hence when new modules are integrated it becomes unfeasible to use structural testing