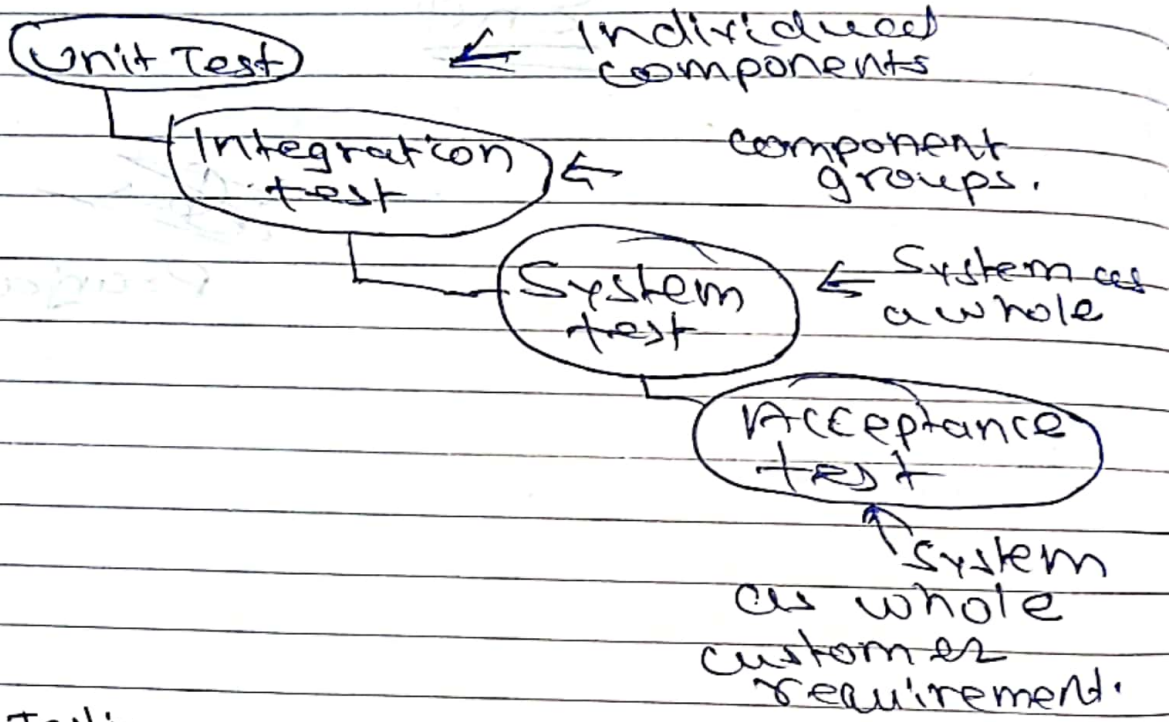
Unit II : Types and Levels of Testing
(weightage : 18 marks)

Q1
2.1 Levels of Testing (2m)



Unit Testing

In this type of testing, error are detected individually from every component or unit by individually testing the component or units of software to ensure that they are fit for use by the developers. It is smallest testable part s/w.

Integration Testing

In the testing, two or more modules which are unit tested are integrated to test.

System Testing

In system testing, complete and integrated software are tested.

Acceptance Testing

This is kind of testing conducted to ensure that the requirements of users are fulfilled before its delivery. software works correctly in users working environment.

Q1 2.2 Unit Testing : Drivers, Stub.

Q2 Describe Unit Testing. (4m) or define (2m) (2m)

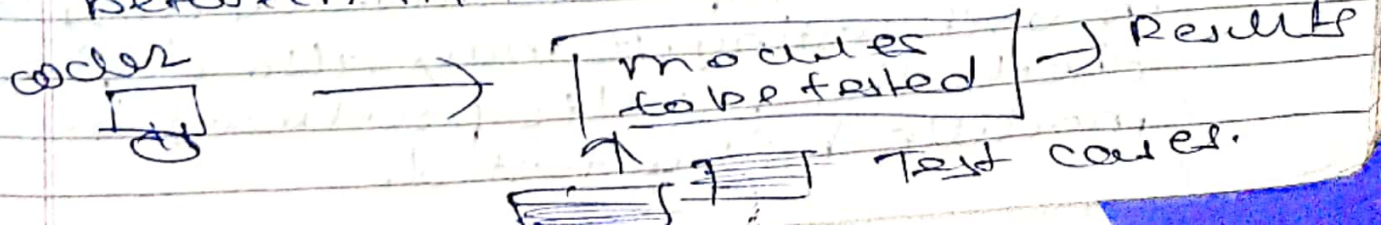
⇒ Testing that occurs at the lowest level is called unit or module Testing.

Unit testing is performed to test the individual units of software.

Unit is smallest part of software system which is testable.

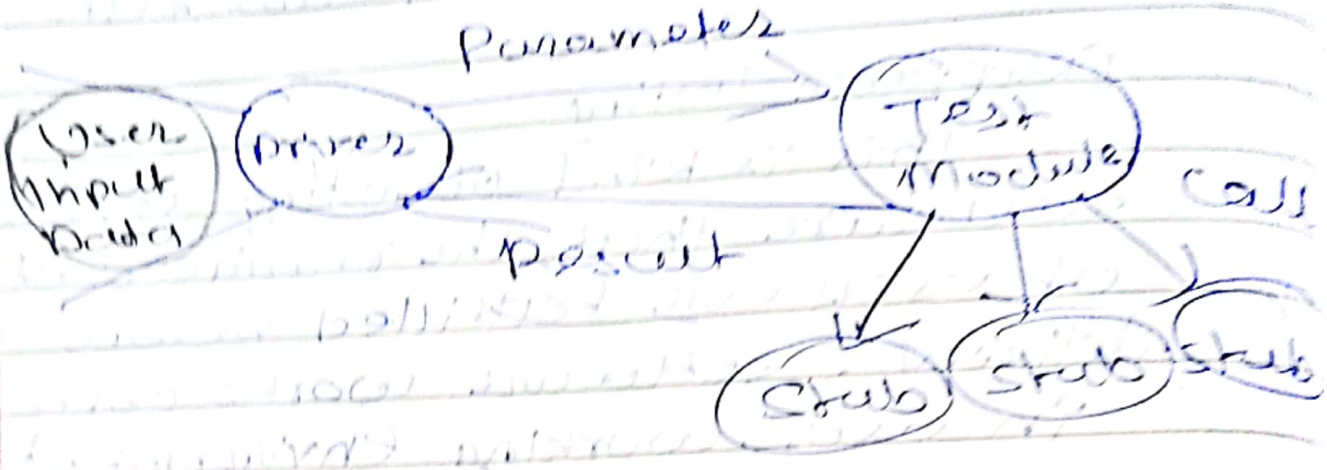
Unit testing may be include code file, classes, method which can be tested individually for correctness.

In Unit Testing each unit is tested seperately before integrating them into modules to test interfaces between modules.



Q3. Describe the concept of drivers and stub with suitable example. (4m)

→



Driver

main program that accepts test case data, passes data to component to be tested and prints/provide relevant results.

Now suppose you have modules A, B, C & modules B & C ready but module A which calls functions from module B & C is not ready so developer write dummy piece of code for module.

A which will returns values to module B and C.

This dummy piece of code is known as Driver.

In simpler terms, A driver module that provide input to module being tested in unit testing. It may also display to check the results produced by module being tested.

Stubs

Subordinate module that are called by the module to be tested.

It is dummy sub-program that does minimal data manipulation, provides verification of entry and returns the control to module "under testing".

Assume you have 3 module. (A, B, C).

Module A is ready and we need to test it, but module A calls functions from module B & C which are not ready.

So developer will write a dummy module which simulates B & C and returns value to module A.

This dummy code is known as Stub.

Important / Need of drivers & Stub.

1) Stubs and Driver replace missing or unavailable modules during testing.
2) They are made with specific function for each module.
3) Developers and testers create stubs & drivers.
4) Used mostly in integration testing
   a) Stubs for top-down testing
   b) Drivers for bottom-up testing.

Q4 Differinicate drivers and stubs. (4M)

Parameter	Stubs	Drivers.
definition	Stubs are dummy modules that always used to simulate low level modules.	Drivers are dummy modules that always used to simulate higher level modules.
usage	Stubs are used in top down approach.	Drivers are used in bottom-up integration.
Purpose	Replacing missing lower-level module	Replacing missing higher-level module.
called as	Stubs are called programs	Drivers are calling called programs.
when to use?	Stubs are used when sub programs are under construction	Drivers are only used when main programs are under construction.
Tested first	Top most module	Bottomost/lowest module
Example	A Stub returns dummy data when called.	A driver calls module to test it.

2.3 Integration Testing

Q5 Illustrate process of bidirectional integration testing. State two advantages and disadvantages. (4m)

or

Q5 Describe the types of integration testing with neat diagram. (4m).

or

State & describe top-down approach of integration testing with diagram. (4m

or

State any two example of Integration testing. (2marks)

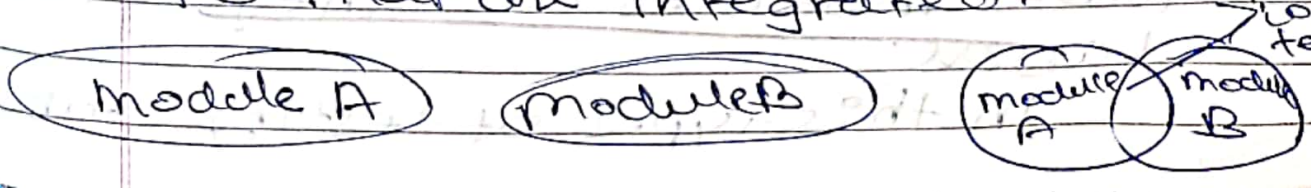
⇒ **Integration testing**

Integration testing process of testing the interface between two software units or modules.

It focuses on determining the correctness of the interface.

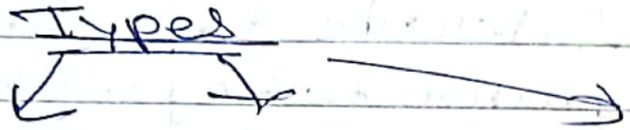
Upon completion of unit testing, the units or modules are to be integrated which gives raise to integration testing.

The purpose of integration testing is to verify functional, performance and reliability between the modules that are integrated.



Objectives of integration testing

- To reduce risk.
- To verify whether functional and non-functional behaviours of interfaces are designed and specified.
- Built confidence in quality of interfaces.



Top-down Integration | Bottom-UP Integration | Sandwich / Bidirectional Integration.

① Top-Down Integration testing

Top-down integration testing, takes place from top to bottom.

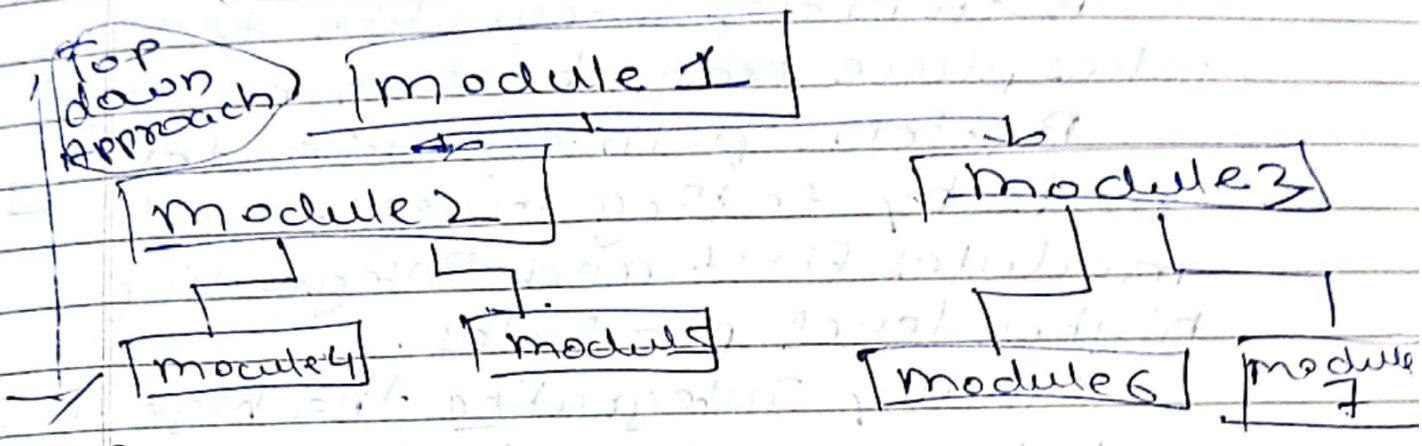
Top-down integration testing is method where testing starts from top (high-level modules) and moves down to lower-level modules.

Finally integrating the low-level module to a high level to ensure the system is working as intended.

First the top module is tested independently, Then step-by-step, lower modules are integrated in Layers.

In this types of testing Stubs

are used as temporary modules if a module is not ready for integration testing especially lower level module.



Diagram

Advantage

- Early detection of major design issues.
- Critical high level functionality is tested first.
- Progress is visible.
- Easier to test UI or main system flows.

Disadvantages

- Stubs may introduce temporary complexity.
- Lower modules are tested later, delaying bug discovery

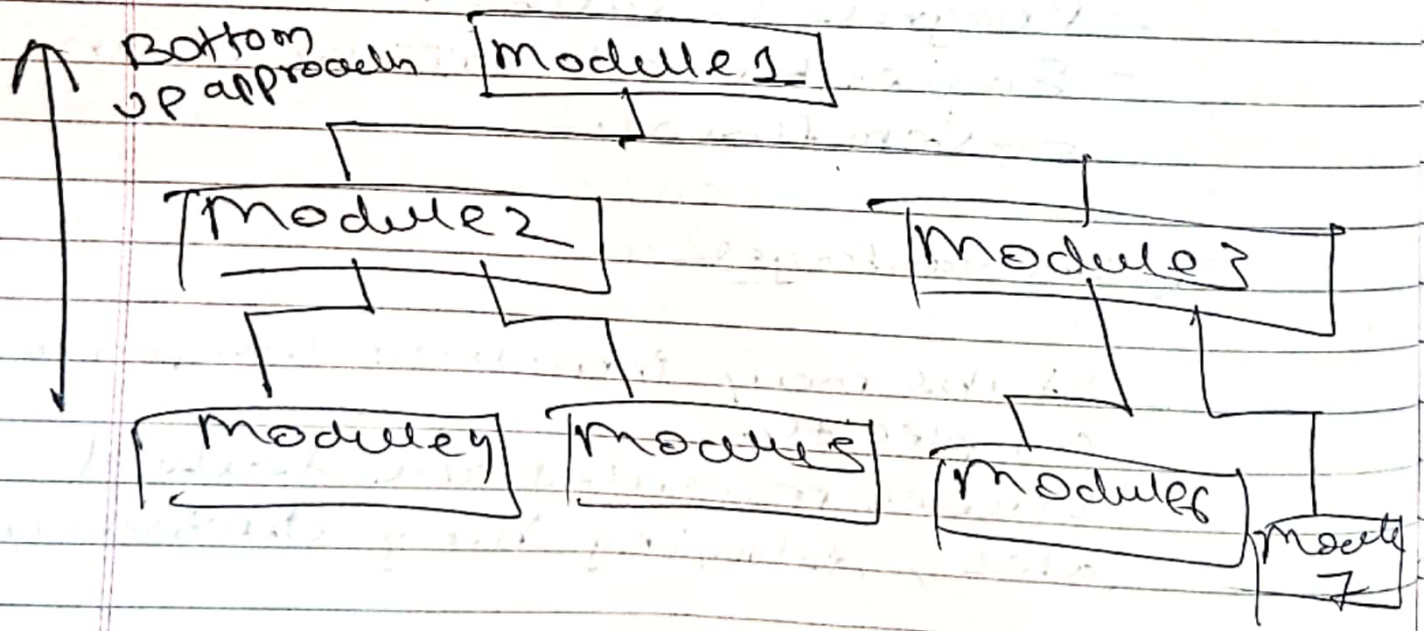
2) Bottom-up Integration

- It is reciprocate of Top-Down Approach.
- Bottom-up Integration testing takes place from bottom to up.

Bottom up Integration testing starts by testing lowers-level modules first and integrating higher level modules.

Finally integrating the high-level modules to a lower level to ensure the system is working as intended.

Drivers are used as a temporary module for integration testing. (simulates higher level modules that are not developed yet).



Advantages

- low level functionality is tested early.
- No need for stubs, as testing starts with fully implemented modules
- Ensures thorough testing of lower-level details and functionality.

Disadvantages

- Requires creation of drivers, which add complexity
- Full system functionality not be tested until late in process.

3) Bi-directional Integration

Bi directional or Hybrid integration testing is also known as Sandwich Integration testing.

It is a combination of both Top-down and Bottom-up integration testing.

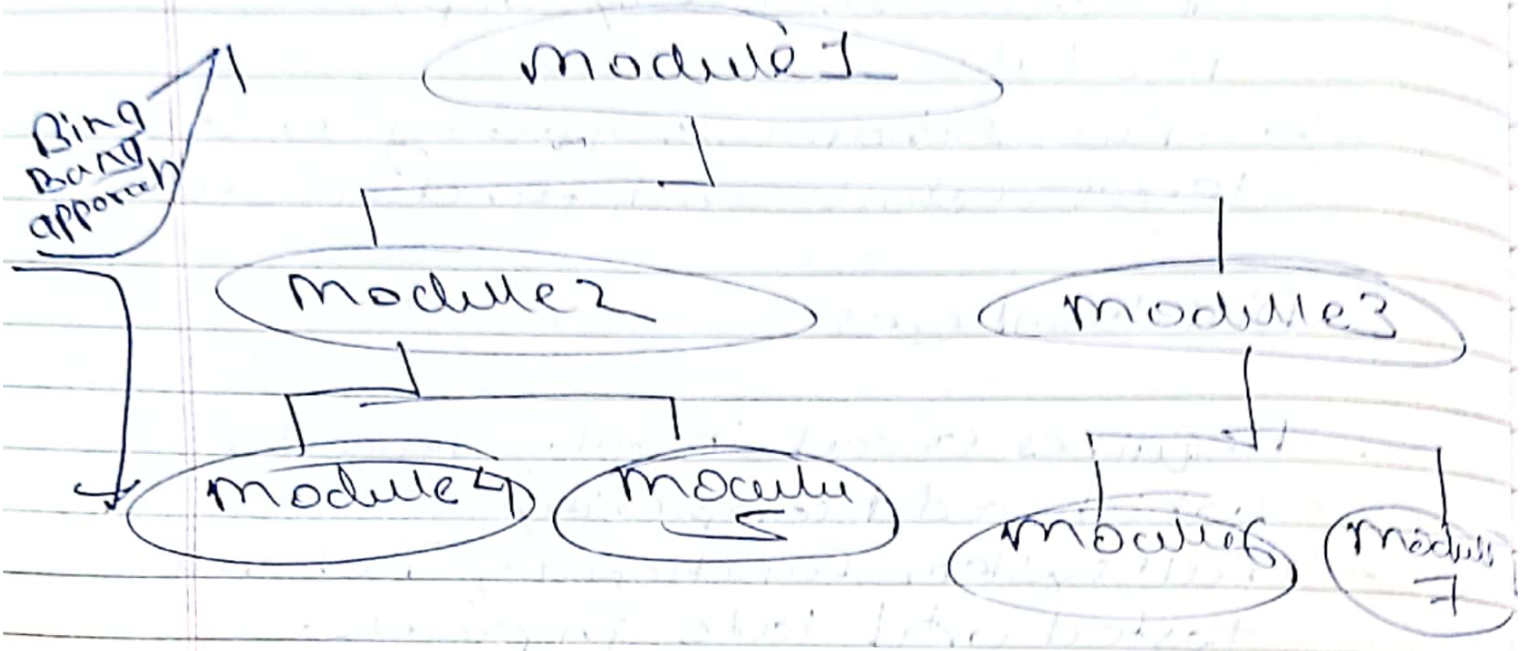
Here, top modules are tested with lower once. At same time, lower modules are tested with top ones.

Stubs and Drivers both are used in this type of testing.

The middle part of system is

testing all together using Bing-Bang approach.

Diagram



Advantages

- Both Top-Down and Bottom up approach ~~of~~ ensure that entire system is tested throughly
- Early Detection of Issues.
- Balanced Approach.
- Effective for Large Projects.
- Parallel Development.

DisAdvantages

- Time Consuming
- Complexity Setup
- Challenging Middle Layers
- High Resource Usage

2.4 : Testing on Web Applications

Q6 Describe with on example each.
i) Load testing
ii) Stress testing (6m/4m).

OR

Explain testing approaches that are considered during web application testing. (4m).

⇒ Note : Performance Testing.
Performance testing is designed to test the run-time performance of s/w within context of an integrated system

1) Load Testing
i) Load testing checks how a system handles many users at one time like in real life.
ii) Testers wants to see how the system responds, find issues, and discover its user capacity.
iii) It helps finding problems like slow response time, bottlenecks, code issues, and memory leaks.
iv) In simpler terms, Load testing is the way to check how a system works when a lot of people or tasks are using at same time.
v) The goal of it is to see the system can handle heavy load, like

many users or action happening at once, without slowing or breaking down.

Working

1) You slowly add users or tasks to system and watch how it reacts.
2) It's done in planned environment so you can compare results between system.
3) The system is pushed to its maximum to see where it might start having problems.
4) The test is successful if everything works properly without errors within expected time.
5) Some systems need to be tested over a long time to make sure they don't slow down or crash.
6) Automated tools like (Jmeter) help to make testing easies and relicable.

Examples

- Downloading big files from internet.
- Running multiple apps at computer once.
- Sending lot of print jobs to printer.
- Constanly reading and writing data to hard drive.

Advantage

It makes fixing problems cheaper.
Finds problems & mistake in system.

(2) Stress testing

i) Stress testing is type of software testing that verifies the stability and reliability of system.

ii) Stress testing is where a system is pushed beyond its normal limits to see how it behaves.

iii) It is non-functional testing. i,e checks how stable a system is pushing it harder than usual.

iv) It tests a system by pushing it to its break point to see if it can handle extreme conditions without crashing.

v) Instead of looking at how the system behaves normally, stress testing checks how the system deals with problems like heavy loads, low memory or lack of resources.

vi) Stress testing is sometimes called Fatigue testing.

Example: Imagine a website handles many users at once. Normally, it works fine with moderate no. of users. But during stress testing, you overload the site with many more users than it built to handle.

This helps check if website slows down, crashes, or loses data when under extreme traffic.

Q. Compare Load & Stress Testing. (Continued)

Load Testing	Stress Testing
Checks how well the system works under heavy use.	Finds the point where the system breaks.
Simulates regular user activity	Floods the system with many request at once.
Used for websites & apps.	Tests how the system handle sudden spikes in use
Helps avoid costly failure.	Helps reduce recovery time after failure.
Measures speed, capacity & performance	Measures stability and response time.
Huge numbers of users.	There are too many users and too much data
Performed to find upper limit of system.	Performed to find behaviours of system.
Example Downloading series of large files from internet.	Example: Flooding server with useless email messages.

Q8 With the help of Diagram, describe client-server Testing. (6m)

or

Explain client-server testing with suitable Diagram. (6M)

Client-Server Testing

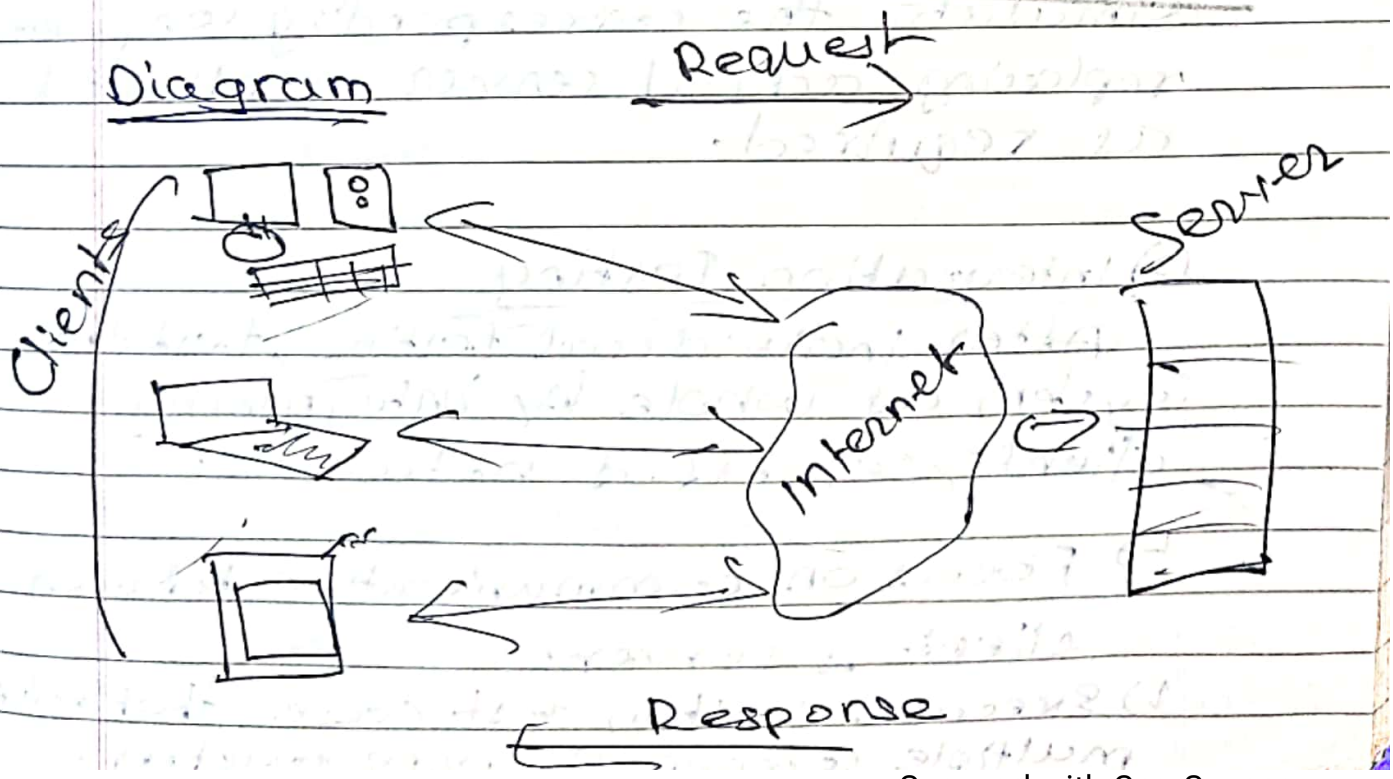
① The client-server application consist of two system, one is client and other side is server.

② Here, client and server intercept with each other over computer Networks.

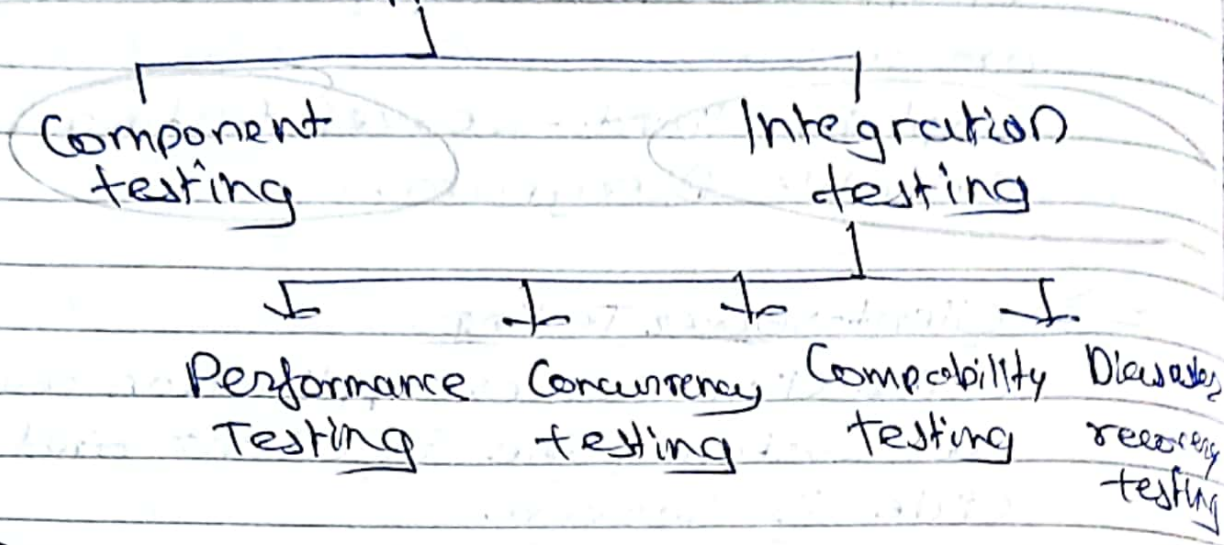
③ Client-server application testing, the client sends request to server for specific information and server sends response back to client with requested information.

④ Hence, this testing is also known as two-tier application testing.

Diagram



Client-server approaches.



① Component - Testing

Test client and server components individually.

Client - Testing: Use a server simulator to test client functionality.

Server - Testing: Use a client simulator to test servers functionality behaviour.

Simulate the corresponding component replacing actual server or client as required.

② Integration Testing

After individual testing, test the system as whole by integrating client, servers & networks.

↳ Focus on communication between client & server.

↳ Execute system test case that involve multiple components work together.

③ Performance testing

- Evaluate system under loads.
- Simulate maximum load (eg many no. of clients communicating with server simultaneously).

④ Concurrency Testing

Test the system's behaviour when multiple user access the same resource or data at a time.

- Check for data consistency & correctness during simultaneous access.
- Simulate scenarios where multiple users access shared records or resources.

⑤ Disaster Recovery Testing

Test how system behaves in case of a failure.

Simulates failures of client, server, or communication link various point.

⑥ Compability testing

Ensures system compability in varying hardware & software environment.

Checks functionality of client, hardware platform, OS & software environments.

Expected

Page No.
Date / /

Q. Security Testing (2M)

Security Testing is testing technique to determine if an information system protects data and maintain functionality as intended.

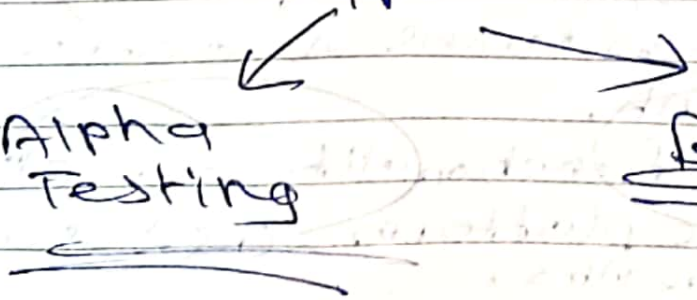
It also aims for 6 basic security principles
1) Confidentiality.
2) Integrity.
3) Authentication
4) Authorization
5) Availability.
6) Non-repudiation.

Q// Acceptance Testing (2M).

Acceptance testing, is a testing technique performed to determine whether or not the software system has meet requirement specifications.

The main purpose of this test is to evaluate the system's compliance with business requirements and verify if it is has met the required criteria for delivery to end users.

Types



Q9 Enlist two advantages of performance Testing. (2m).

⇒ ① How well product is developed.
② A product is developed.
③ A product is what actually the customer need.
④ Minimize or eliminate the issues arising from production.

Q10 Explain any two special tests.

or

⇒ ~~Explain~~

Q/ Explain Regression testing.

⇒ Regression Testing is the process used in software testing to ensure that recent changes, modification, such as code, bugs fixes or new features have not introduced new defect into existing functionality.)
The main goal is to verify that software continues to perform correctly after updates, ensuring that previously tested software still work as intended intented.

When to Perform ?

- After fixing bugs.
- After adding new feature
- After performance improvement.
- During regular maintenance.
- Before major releases
- After configuration changes.

Types of Regression Tests:

1) Final Regression Test.
Done when the software hasn't changed for while.
Purpose: to check if it's ready to released or Deployed.

2) Normal Regression Test.
Done after every change.
To check if the recent changes have broken anything else.

Entry Criteria

1) code changes are made.
2) Test case data are prepared.
3) Testing environment is ready & stable.
4) Previous issues are fixed and system is ready for testing.

Exit Criteria

1) All test have been run.
2) No critical bugs remain.
3) Required areas have been tested.
4) Test results are documented and approved by the team.

Q what is GUI Testing? (2M/4M)

⇒ There are two types of interfaces for computer application.

(1) Command Line Interface: it is where you type text and computer responds to the command.

(2) GUI stands for Graphical User Interface where you interact with computer using images/icons rather than text.

GUI Testing

i) GUI Testing process of testing the system Graphical User Interface of Application under Test.

ii) GUI testing involves checking the screen with control like menus, buttons, icons and all types of bar - tott toolbars, menu bars, dialog boxes & windows etc....

iii) GUI is what the user sees. A user does not see the source code. The interface is visible to user.

iv) Especially the focus is only to design structure, image that they are working properly or not.

traits : 1) Check screen validations.
2) Verify All Navigations.
3) Check usability conditions.
4) Verify dateField & Numeric Field formats.

5) Verify Data Integrity of object states.

Q11 Differinate Smoke Testing & Sanity Testing. (4m)

Smoke Testing	Sanity Testing
Smoke testing is done to assure that the cecute functionalities of program is working.	Sanity testing is done to check the bugs have been ~~butld~~ fixed cefter build.
Also known as subset of Acceptance Testing	Also known as subset of Regression Testing.
It is documented	It isn't documented
performed by testers or developers	performed by testers.
may be stable or unstable	It's stable
scripted	not scripted
It's like genral health check up	It's like specalized health check up.
Efficient	less - Efficient
more time	less time

Q2 Compare Alpha & Beta Testing. ★★★

	Alpha Testing	Beta-Testing
	The purpose of alpha testing is to identify & fix bugs in s/w before beta testing.	Gathers real users feedback, find remaing bugs.
Technique used	Uses both white & Black Box testing	Uses Black-Box Testing
performed by	Testers who are internal employees of organisation.	Performed by clients or end-users not an part of orginisation
performed at	Alpha testing is performed at the developers site.	Beta testing is performed at end-user product.
Execution	may require long execution cycle.	Only few weeks of execution.
Test cycles	Multipe test cycles are organized in alpha tesing	Only one or two
Environment	Lab or development environment	Real-word user environment
Types	Open alpha Close alpha	closed Beta Open Beta focused Beta

End