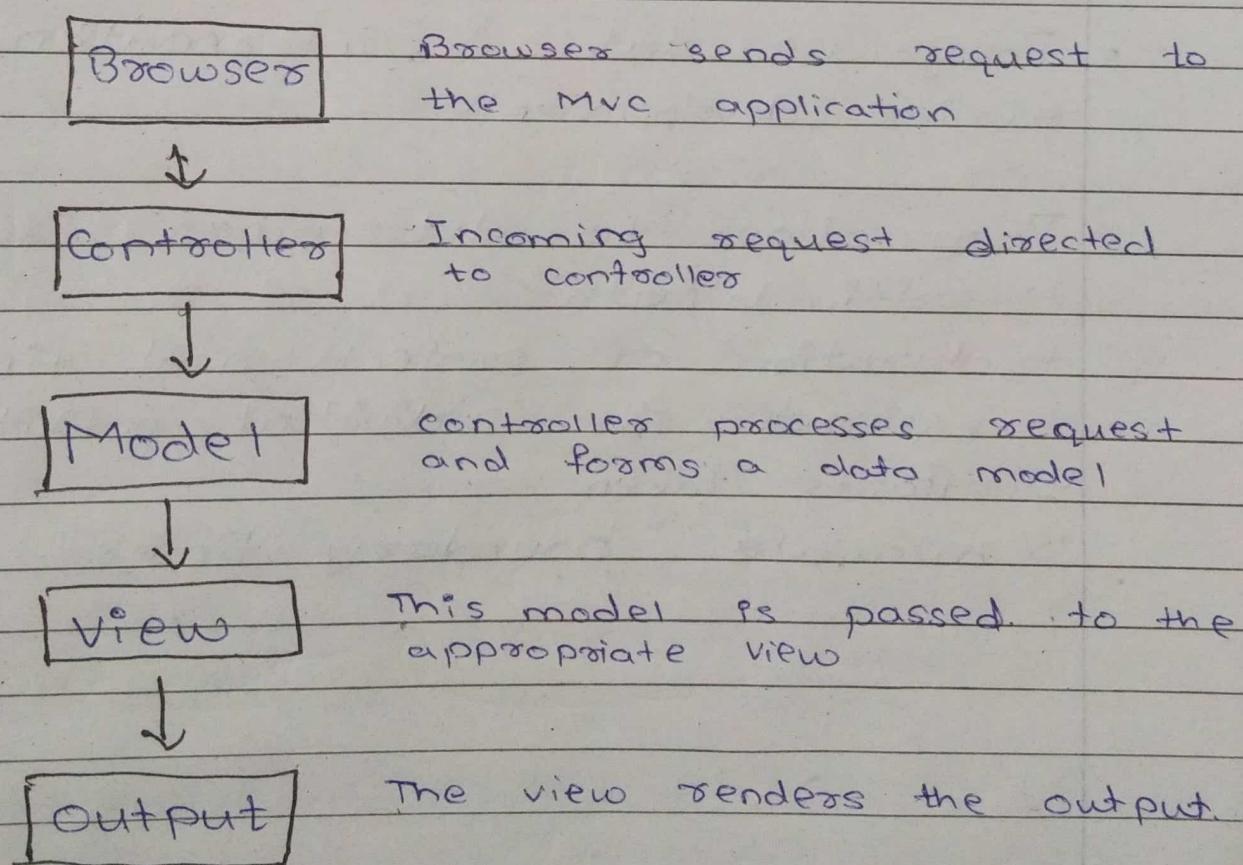


Q) Explain MVC architecture.

- Ans - The Model View Controller (MVC) is an architectural pattern that separates an application into three main logical components:
- ↳ Model
 - ↳ View
 - ↳ Controller.

- MVC takes place in five steps which are given with a flow diagram

• Flow Diagram



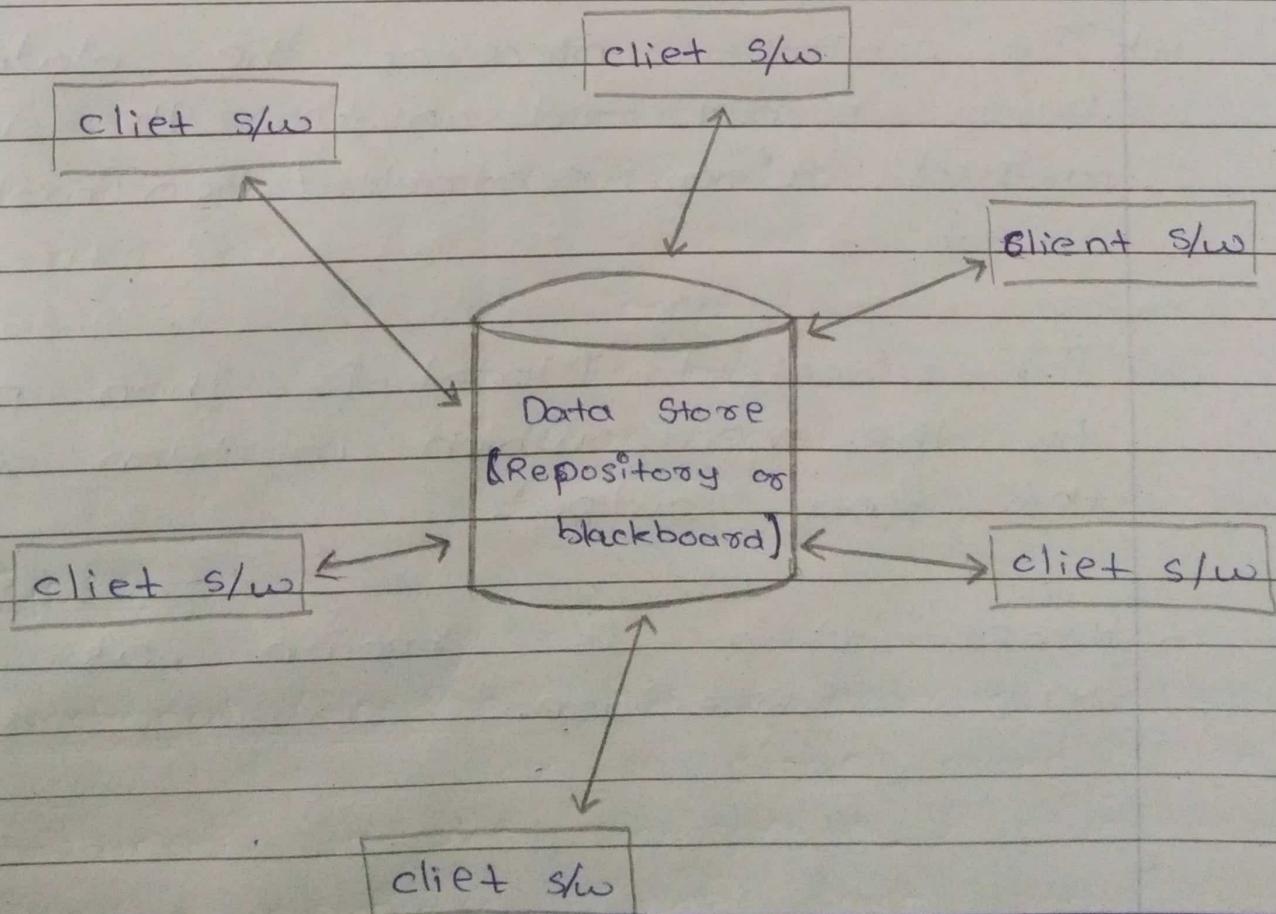
- Flow Steps

- 1) The client browser sends request to the MVC Application.
- 2) Global.asax receives this request and performs routing based on the URI of the incoming request using the RouteTable, RouteData, UrlRoutingModule-Module and MvcRouteHandler objects.
- 3) This routing operation calls the appropriate controller and executes it using the IControllerFactory object and MvcHandler object's Execute method.
- 4) The controller processes the data using Model and invokes the appropriate method using ControllerActionInvoker object.
- 5) The processed Model is then passed to the view, which in turn renders the final output.
 - Hence this is known as MVC architecture.

24. Explain Data Centered Architecture.

- Ans - In data-centered architecture, the data is centralized and accessed frequently by other components, which modify data.
- The main purpose of this style is to achieve integrality of data.
 - It consists of different components that communicate through shared data repositories.
 - The components access a shared data structure and are relatively independent; in that, they interact only through the data store.

- Figure



- Types of components used in milennium

↳ A **central data** structure or data store or data repository, which is responsible for providing permanent data storage. It represents the current data.

↳ A **data accessor** or a collection of independent components that operate on the central data store, perform computations, and might put back the results.

- Interactions or communication between the data accessors is only through the data store.

- The data is the only means of communication among clients.

- The flow of control differentiates the architecture into two categories.

↳ Repository Architecture Style

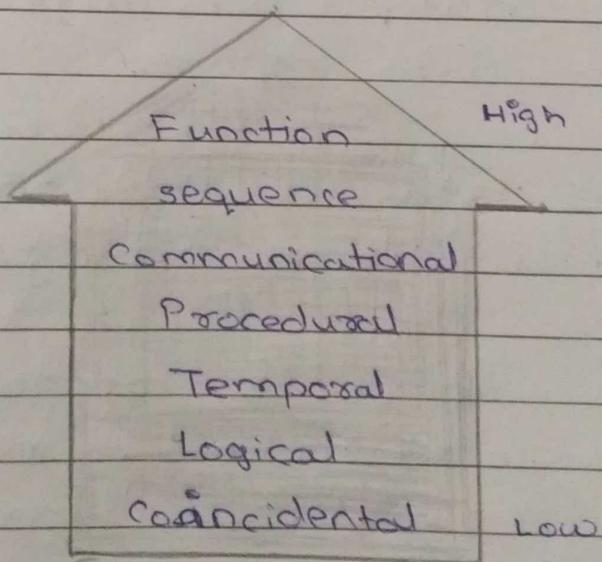
↳ Blackboard Architecture Style.

3) What is cohesion and coupling?

Write down types of cohesion and coupling.

- Ans
- **Cohesion :-** It is a measure of the degree to which the elements of the module are functionally related.
 - It is a degree in which all the elements are dissected to a single task are contained in the component.
 - Basically, cohesion is a glue that keeps the module together.
 - A good software design will have high cohesion.

- **Figure :-**



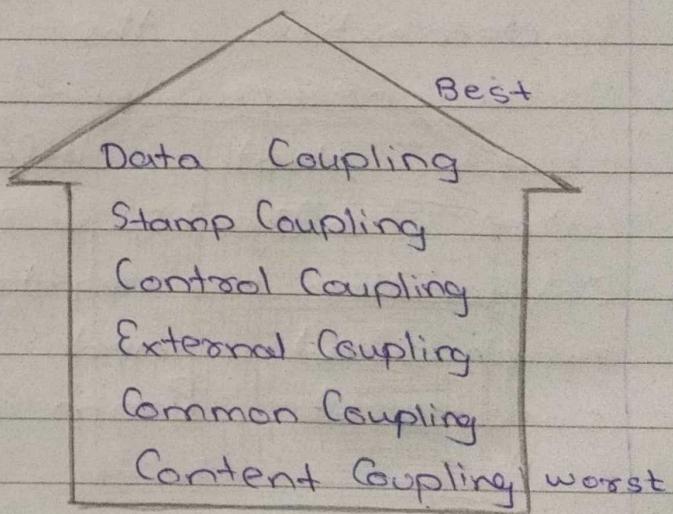
- **Types**

↳ **Functional Cohesion :-** Every essential element for a single computation is contained in the component. A functional cohesion performs the task and functions.

It is an ideal situation.

- ↳ Sequential Cohesion :- An element outputs some data that becomes the input for other element, i.e., data flow between the parts. It occurs naturally in functional programming languages.
- Coupling :- It is the measure of the degree of interdependence between the modules. A good software will have low coupling.

• Figure :-



• Types

- ↳ Stamp Coupling :- The complete data structure is passed from one module to another module. Therefore, it involves stamp data. It may be necessary due to efficiency factors - this choice made by the insightful designer, not a lazy programmer.

- ↳ External Coupling :- The modules depend on other modules, external to the software being developed or to a particular type of hardware.
Ex - protocol, external file, device format, etc.

- ↳ What is Abstraction and Data Hiding.

Ans • Abstraction

- ↳ It is hiding the internal implementation and just highlight the set of services.
- ↳ It is achieved by using the abstract class and interfaces and further implementing the same.
- ↳ Only necessarily characteristics of an object that differentiates it from all other objects.
- ↳ Only the important details are emphasized and the rest all are suppressed from the user or reader.

• Types

- ↳ Procedural Abstraction
- ↳ Data Abstraction
- ↳ Control Abstraction

• Data Hiding

- ↳ It is hiding internal data from outside.

users.

- ↳ The internal data should not go directly that is outside person/classer is not able to access internal data directly.
- ↳ It is achieved by using an access specifier - a private modifier.

5) Explain white & black box testing.

Ans ① White-Box Testing :- It is detailed investigation of internal logic & structure of the code.

- White box testing is also known as glass testing, clear box testing, structural testing, open-box testing, transparent box testing.
- In testing, a tester needs to know the internal workings of the code.
- It is a method of software testing that tests internal structure or workings of an application as opposed to its functionality (i.e. black box testing).
- In testing an internal coding & infrastructure of a software focus on checking of pre-defined inputs against expected & desired outputs.
- It is based on inner workings of an application & revolves around internal structure testing.
- In this type of testing programming

skills are required to design test cases.

- The primary goal of white box testing is to focus on the flow of inputs & outputs through the software & strengthening the security of the software.
- The white-box testing contains various tests.
 - Path testing
 - Loop testing
 - Condition testing
 - Testing based on the memory perspective
 - Test performance of the program.
- o Reasons for white-box testing:

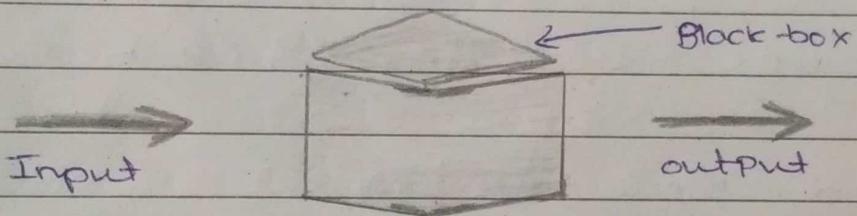
- It identifies internal security holes
- To check the way of input inside the code
- check the functionality of conditional loops
- To test functions, object & statement at an individual level.

- Advantages :- It optimizes code so hidden errors can be identified.
- Test cases of white-box testing can be easily automated.
- It can be started in the src phase even without GUT

- :- Disadvantages :- It is much expensive & complex.
- It can lead to production errors because it is not developed by the developers.
- It is too much time consuming when it comes to large-scale programming applications.
- It needs professional programmers who have a detailed knowledge & understanding of programming language & implementation.
- Techniques used in white-box testing are
 - ↳ Data Flow testing
 - ↳ Control Flow testing
 - ↳ Branch Testing
 - ↳ Statement Testing
 - ↳ Decision Testing

- Black-Box Testing :- The techniques of testing without having any knowledge of the interior workings of the application is called black-box testing.
- It is a technique of software testing which examines the functionality of software without peering into its internal coding or structure.
- The primary source of black-box testing is a specification of requirement that is stated by the customer.

- The tester is oblivious to the system architecture & does not have access to the source code.
- In this method, tester selects a Fn & gives input value to examine its functionality & checks whether the function is giving expected output or not.
- If the function produces correct output then it is passed in testing, otherwise Failed.



- Techniques used in black-box testing are
 - ↳ Decision table technique
 - ↳ Boundary value technique
 - ↳ State transition technique
 - ↳ All-pairs testing technique
 - ↳ Cause-Effect technique
 - ↳ Equivalence Partitioning technique
 - ↳ Error guessing
 - ↳ Use case technique
- Generic steps of black-box testing:

Step-1) The black-box test is based on the specification of requirements, so it is

examined in the beginning

- step 2] The tester creates a positive test scenario & an adverse test scenario by selecting valid & invalid input values to check that the software is processing them correctly or not.
- step 3] The tester develops various test cases such as decision table, all pair-test, error estimation, cover-effect, etc.
- step 4) The execution of all test cases.
- step 5) The tester compares the expected output against the actual output.
- step 6) If there is any flaw in the software, then it is revised & tested again.

- Advantages :- Well suited & efficient for large code segments.
- code access is not required.
- clearly separates user's perspective from the developer's perspective through visibly defined roles.
- Large numbers of moderately skilled testers can test the application with no knowledge of implementation programming, language or operating systems.
- Disadvantages :- Limited coverage since only a selected number of test scenarios is actually performed.

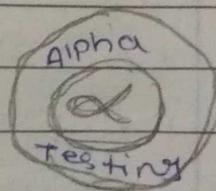
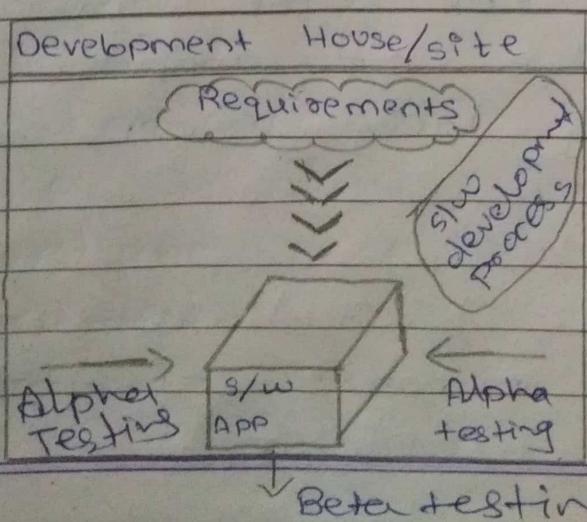
- The test cases are difficult to design.
- In efficient testing, due to the fact that the tester only has limited knowledge about an application.

6. Explain Alpha & beta Testing.

Ans

Alpha Testing :- It is conducted in the organization & tested by a representative group of end-users at the developer's side & sometimes by an independent team of testers.

- It is used to identify all issues & bugs in software before releasing of the software.
- The black box & white-box techniques are used in alpha testing.
- It can be depends on the requirements particular lab environment & simulation of the actual environment required for this testing.
- It is simulated or real operational testing at an in-house site.



- Alpha testing follows the following process.

- 1) Requirement Review :- Review the design of the specification & functional requirement.

- 2) Test Development :- It is done on the outcome of the requirement review develop the test cases & test plan.

- 3) Test case design :- Execute the test plan & test cases.

- 4) Logging defects :- Logging the identified & detected bug found in the application.

- 5) Bug Fixation :- When all the bugs are identified & logged, then there is a need to fix the bug.

- 6) Retesting :- When all the issues are solved & fixed retesting is done.

- Alpha testing ensures that the software performs flawlessly & does not impact the expectation of the organization; the company implements final testing in the form of alpha testing.

- Reasons for alpha testing are :-
 - ↳ It allows the team to test the software in a real-world environment
 - ↳ It ensure the success of the software product.
 - ↳ Refines the software product by finding & rectifying bugs that weren't discovered through previous tests
- Features :- It is a type of acceptance testing.
 - It is in the labs where we provide a specific & controlled environment.
 - There is not any involvement public acceptance of the software product
 - It is done often the unit testing, integrator testing, system testing but before the beta testing
- Advantages :- It provides a complete test plan & test cases
 - Free the team members for another project.
 - Every feedback helps to improve software quality.
 - It reduces the delivery time of the project.

- Disadvantages :- It does not involve in-depth testing of the software.
- The lab environment is used to simulate the real environment.
- The difference between the tester's tests & the customer's data from their perspective may result in the discrepancy in the software functioning.

- Beta Testing :- It is performed by real users of the software application in a real environment.
- It is a type of User Acceptance Testing, among the most crucial testing, which performed before the release of the s/w.
 - It is a type of field test.
 - This testing performs at the end of the software testing life cycle.
 - It is the last phase of the testing, which is carried out at the client's site.
 - It helps in minimization of product failure risks & it provides increased quality of the product through customer validation.

- Types of beta testing:-

- ↳ Open Beta testing
- ↳ Closed Beta testing
- ↳ Traditional Beta testing
- ↳ Public Beta testing
- ↳ Technical Beta testing
- ↳ Focused Beta testing
- ↳ Post-Release Beta testing.

- Criteris for Beta Testing:-

- Entry Criteria :-

- ↳ Sign off the document from alpha testing.
- ↳ Beta version of the s/w should ready.
- ↳ The environment should be ready to release the software application to the public.
- ↳ To capture the seat-time faults environments should be ready.

- Exit Criteria :-

- ↳ All the major & minor issues resolved.
- ↳ The feedback report should prepare
- ↳ The delivery of beta test summary report.

- Advantages :- It focuses on the customer's satisfaction.
- It helps to reduce the risk of product failure via user validation.
- It helps to get direct feedback from users.
- It helps to detect the defect & issues in the system, which is overlooked & undetected by the team of software testers.

- Disadvantages :- It can be a time-consuming process & can delay the final release of the product.
- A software engineer has no control over the process of the testing, as the users in the real-world environment perform it.

7) What is code review.

- Ans - Code review is a systematic examination, which can find & remove the vulnerabilities in the code such as memory leaks & buffer overflows.
- It is a software quality assurance activity in which one or several people check a program mainly by viewing & reading parts of its source code & they do so after implementation or as an interruption of implementation.

- The persons performing the checking, excluding the author are called reviewers.
- Technical reviews are well documented & use a well-defined defect detection process that includes peers & technical experts
- It is ideally led by a trained moderator who is NOT the author
- This kind of review is usually performed as a peer review without management participation.

- Types :-

- ↳ Management review
- ↳ Technical reviews
- ↳ Inspections
- ↳ Walk-through
- ↳ Audits

Q) Explain system testing & what is smoke test.

Ans ① System Testing :- It is type of software testing or black box testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements.

- It is end-to-end testing where the testing environment is similar to the production environment.
 - It is system testing, integration testing passed components are taken as input. The goal of integration testing is to detect any irregularity b/w the units that are integrated together.
 - System testing detects defects within both the integrated units & the whole system.
 - The result of system testing is the observed behaviour of component or a system when it is tested.
 - It is carried out on the whole system in the context of either system requirement specifications or functional requirement specification or in the context of both.
 - It is basically performed by a testing team that is independent of the development team that helps to the quality of the system impartial.
 - It has both functional & non-functional testing.
- System testing is performed in the following steps:
- Test Environment setup
 - Create Test case

- Create Test Data
- Execute Test case
- Defect Reporting
- Regression Testing
- Log Defects
- Retest.

- Types of system Testing :-

- 1) Regression testing
- 2) Food testing
- 3) Functional testing
- 4) Recovery testing
- 5) Migration testing
- 6) Usability testing
- 7) Software & hardware testing
- 8) Performance testing
- 9) Stress testing
- 10) Reliability
- 11) Regulatory & compliance
- 12) Scalability testing

- Importance of system Testing :-

- ↳ It gives 100% assurance of system performance as it covers end-to-end function of the system.
- ↳ It includes testing of system software architecture & business requirements.
- ↳ It helps in mitigating live issues & bugs even after production.

- It uses both existing system & a new system to feed same data in both & then compare the difference in functionalities of added & existing functions so, the user can understand benefits of new added functions of the system.

- Advantages :-

- ↳ It increases the confidence level of the team in the product before the product goes for acceptance testing.
- ↳ It helps in getting maximum bugs before acceptance testing.
- ↳ It is a black box testing hence testers do not need programming knowledge to perform it.

② Smoke Testing :- It is also known as confidence testing, Build verification testing, build acceptance test.

- It is a testing technique that is inspired from hardware testing, which checks for the smoke from the hardware components once the hardware's power is switched on.

- It is software testing method that determines whether the employed build is stable or not.
 - It is a minimum set of tests run on each build.
 - It ensure that the product is testable.
 - We do smoke testing to make sure that the application is installed correctly.
-
- Types of smoke testing :-
 - 1) Formal smoke testing
 - 2) Informal smoke testing
 - Example :- E-Commerce Site.