

Unit III: Software Modelling and Design.

(Weightage - 14 marks)

① Describe and Explain concept of White Box Testing and Black Box Testing. (4M/6m)

⇒ Testing is defined as "Execution of a work product with intent to find a defect".

In simpler words, testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

Testing methods

Once, the software is developed it should be tested in a proper manner before the system is delivered to the user.

For this, two techniques that provide system guidance for designing tests are used.

Testing method

white-Box
Testing

Black-Box
Testing

Black-Box Testing

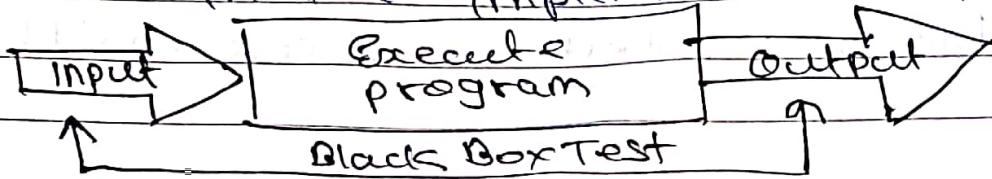
Once, the specified functions of for which the software has been designed is known, test are performed to ensure that each function is working properly.

This is referred to as Black Box testing

- Black Box (or functional) testing checking the functional requirements and examines the input and output data of these requirements.
- Black Box testing is also known as Behavioral testing.
- It is a software testing method in which the internal structure/design/implementation of item being tested is known to the tester.
- In Black Box testing the tester is concentrating on what the software does, now how it does it.
- When Black Box testing is performed, only the legal sets of 'legal' input and corresponding outputs should be known to be to tester and not the internal logic of the program to produce that output.
- Hence, to determine the functionality, the outputs produced for the given sets of input are observed.

Purpose of Black Box testing

- To test functionality of software.
- Address validity, behaviour and performance of software.
- To ensure all components of software that are implemented tested.



White Box Testing

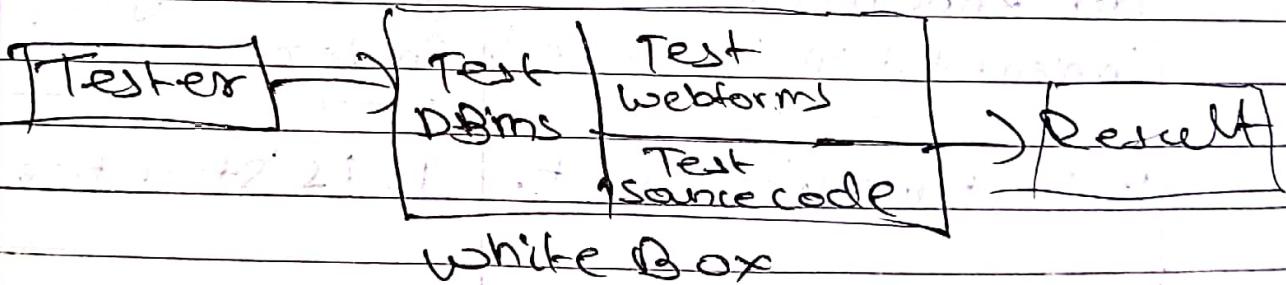
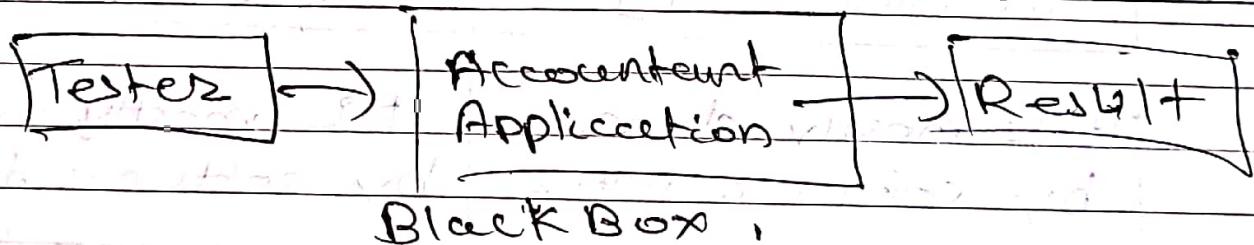
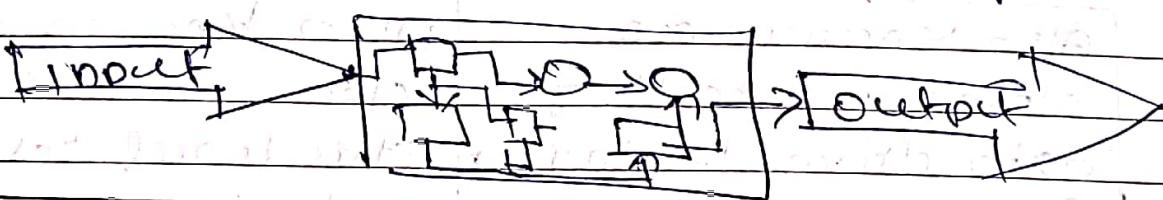
Once, the internal working of software is known, tests are performed to ensure that all internal operations of software are performed according to specification.

This is referred as White Box Testing.

- white box testing is also called Structural Testing and Glass Box Testing.
- white box is a testing technique that takes into account the internal mechanism of a system or component.
- In this testing, test the artefact from internal point of view internal structure of application.
- Here the testers require knowledge of how software is implemented, how it works.
- It needs more structural understanding and knowledge of software structure and implementation.
- White Box test cases cannot be started in initial phase of project because it needs more architecture clarity which is not available at start of project.
- These test cases are written after Black Box test cases are written.

Purpose, White - Box Testing

- 1) To test internal structure of software.
- 2) Address flow and control structure of program.
- 3) All requirements complete implementation.



③ Difference Between White Box and Black Box, (6M) (H.M)

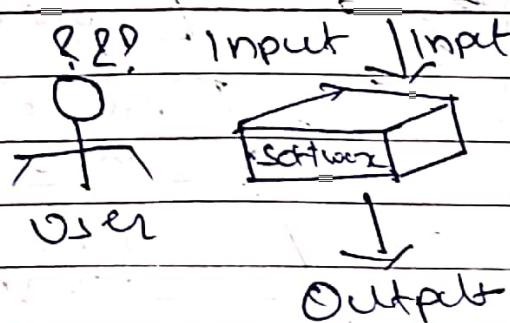
<u>Black Box Testing</u>	<u>White Box Testing</u>
The internal working of an application need not to be known.	Tester have full knowledge of internal working of the application.
Also known as closed-box testing, data-drive testing or functional testing.	Also known as clear-box testing, structural testing or code-based testing.
It is mostly done by software testers.	It is mostly done by software developers.
No knowledge of implementation is needed.	Knowledge of implementation is required.
It is functional test.	It is structure test.
No knowledge of programming required.	Mandatory to know knowledge of programming.
Not suited for algorithm testing.	Suited for algorithm testing.
It is least time consuming.	It is most time consuming.
It is the behavior testing of Software.	It is logic testing of Software.

It is applicable to higher levels of testing of software.

Types

functional Non functional Regression

Less exhaustive

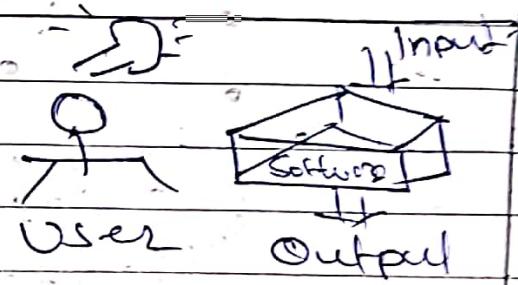


It is applicable to lower levels of testing of software.

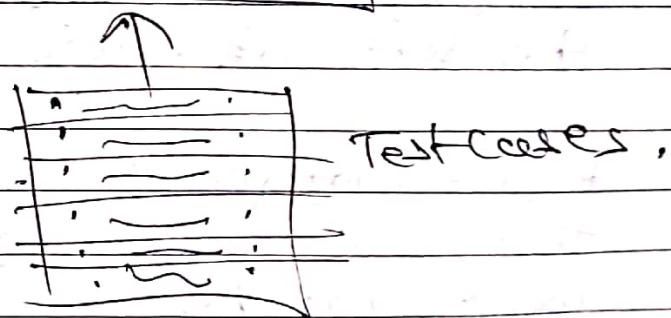
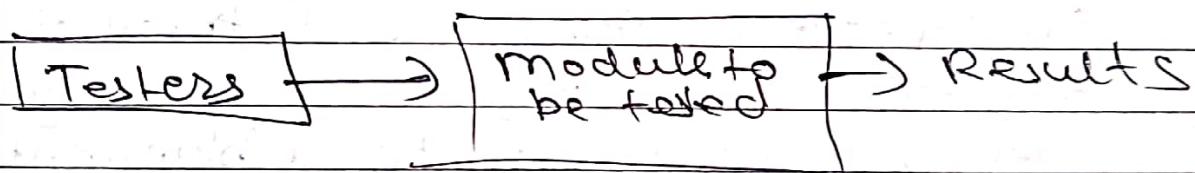
Types

path loop conditinal

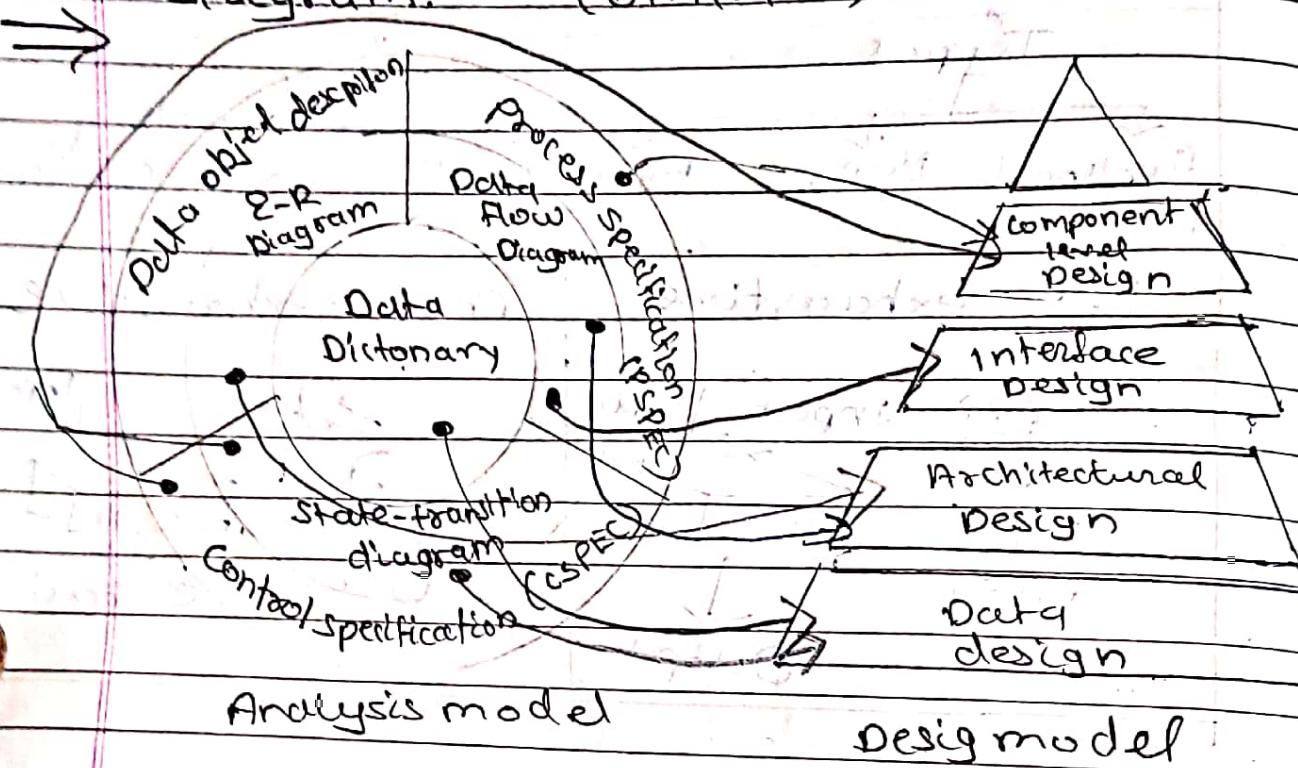
more exhaustive



Unit Testing



~~Imp~~ ③ Translating the requirements model into Design Model with Suitable Diagram. (6m(4m)).



In simpler terms, the process of translating requirements into design involves using different design model to create a blueprint for the software. Here it how work's.

Requirement Model

This is where we gather all the information about what the software should do, including scenarios, classes, flows and behaviors.

Design Model

We use various design methods and notations to create four main design models:

- Data/Class Design

This is model defines how the classes identified in the requirements will actually work in the software and what data structures they'll need.

- Architectural Design

Here we define how different part of the software will be organized and how they'll interact. We also consider architectural styles and patterns to meet the requirements.

- Interface Design

This is about how the software will communicate with other systems and with users. We design how data and control will flow through the software and what kind of behaviour it will exhibit.

- Component Design

This model breaks down the software's architectural into detail descriptions of individual components.

We use information from class-based, flow and behavioral model for this.

Flow of Information

The information from the requirements model feeds into each of these design tasks.

For example, class structures and data content help with data / class design, while scenarios and behavioral models are useful to interface design.

Decision making

Throughout the design process, we make decisions that will impact how the software is built and maintained.

This includes choices about how classes will be realized, how the architectural will be structured, how interface will function, and how components will be designed.

Importance Of Design

Good design is crucial because it lays the groundwork for successful software construction and future maintenance.

By carefully translating requirements into well-thought-out design model, we can ensure that software meets its goals effectively and can be easily managed and updated in the future.

④ Explain the notations used for preparing data flow diagram. (4m).

Symbol	Notation	Use
External Entity		External entity are objects outside the system
Process		A process receives input data and process output data with different form
Data flow		Dataflow is path for data to move from one end to other
Data store		Data stores are repositories for data.

Circle: A circle bubble shows a process that transforms data inputs into data output.

Data flow: A curved line shows the flow of data into or out of process or data store.

Data store: A set of parallel lines shows a place for the collections of data items. A data store indicates that the data is stored which can be used at a larger/later stage or by the order other processes in different order.

The data store can have an element or group of elements.

Source or Sink: Source or Sink is a external entity and acts as a source of system inputs or sink of systems outputs.

Q) Define data objects, attributes, relationship, cardinality with example each. (6marks)

⇒ Data object

A data object is an entity/object in the real world with an independent existence that can be differentiated from other object

A region of storage that contains a value or group of values.

Example, An entity might be

- An object with physical existence (eg. Student, car)
- An conceptual existence (eg. a course job)



Attributes

Each data object/entity is described by a set of attributes.

Example, Employee = Name, Address, Birthdate(Age), Salary)

Each attribute have a name and is associated with an entity and a domain of legal values.

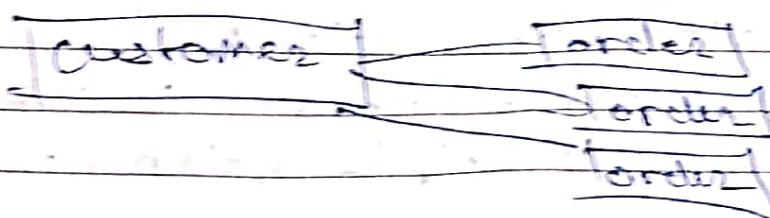
Relationship

- A relationship identifies names and its defines an association between two entity types. One to One relationship.
- Example: We have relationship between the Customer table and Address table.
- If each address table can belong to only one customer, that relationship is "One to One".



One to many

Each customer may have zero, one or multiple orders. But an order can belong to only one customer.

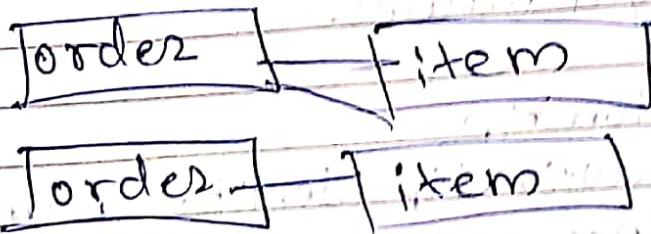
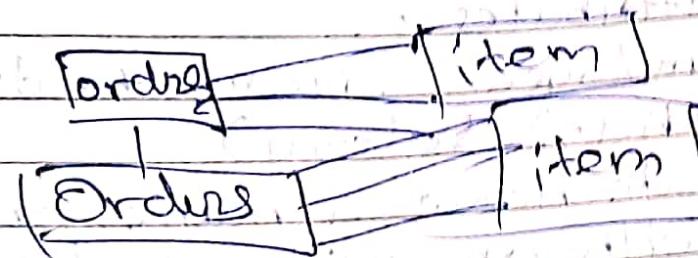


Normal

Page No. 1
Date 1/1

Many to Many

Each order can have multiple items.



Cardinality

In the case of Data Modelling,

Cardinality defines the number of attributes in one entity set, which can be associated with the number of attributes of other set via via relationship set.

Example

one - to - one

One - to - many

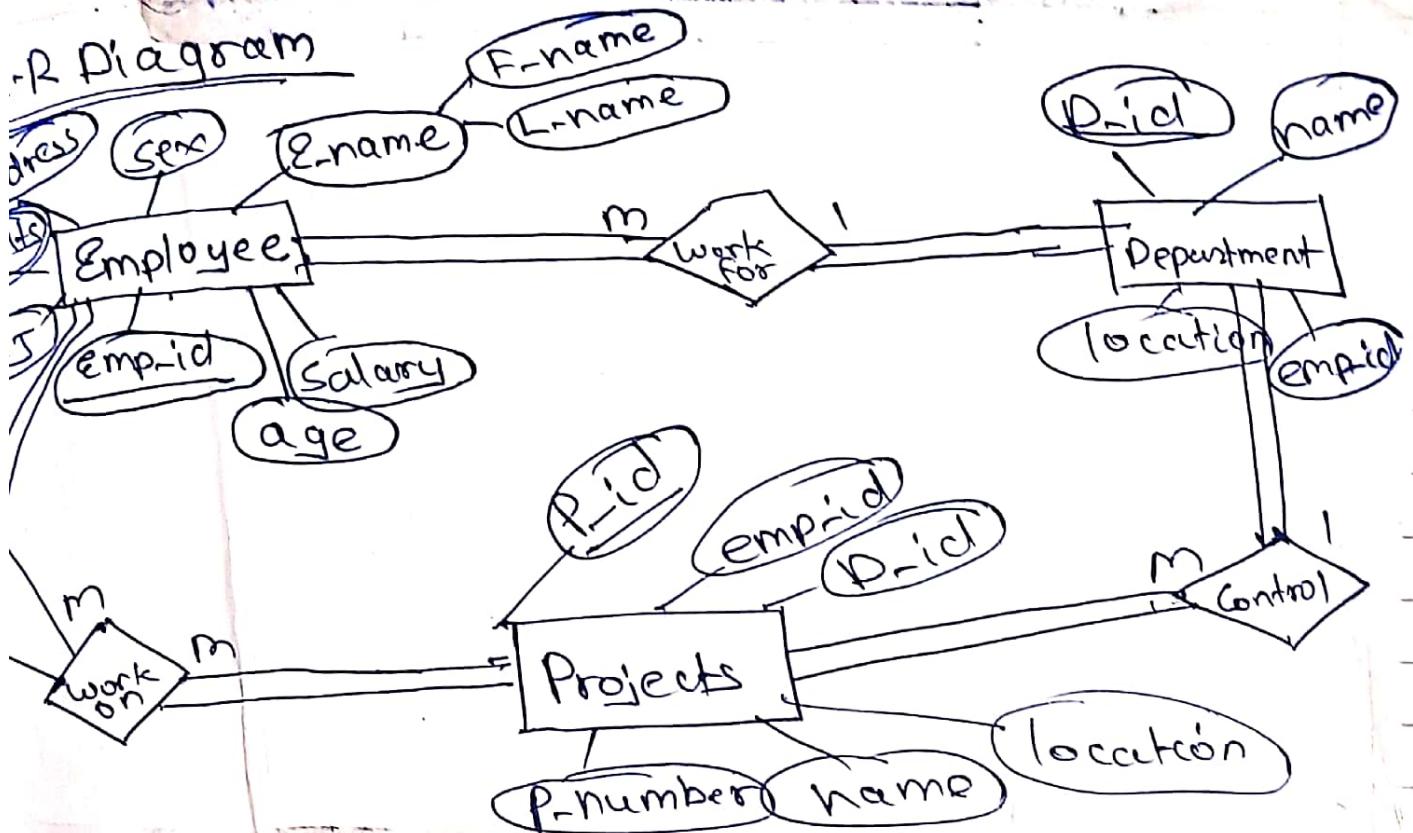
many - to - one

many - to - many

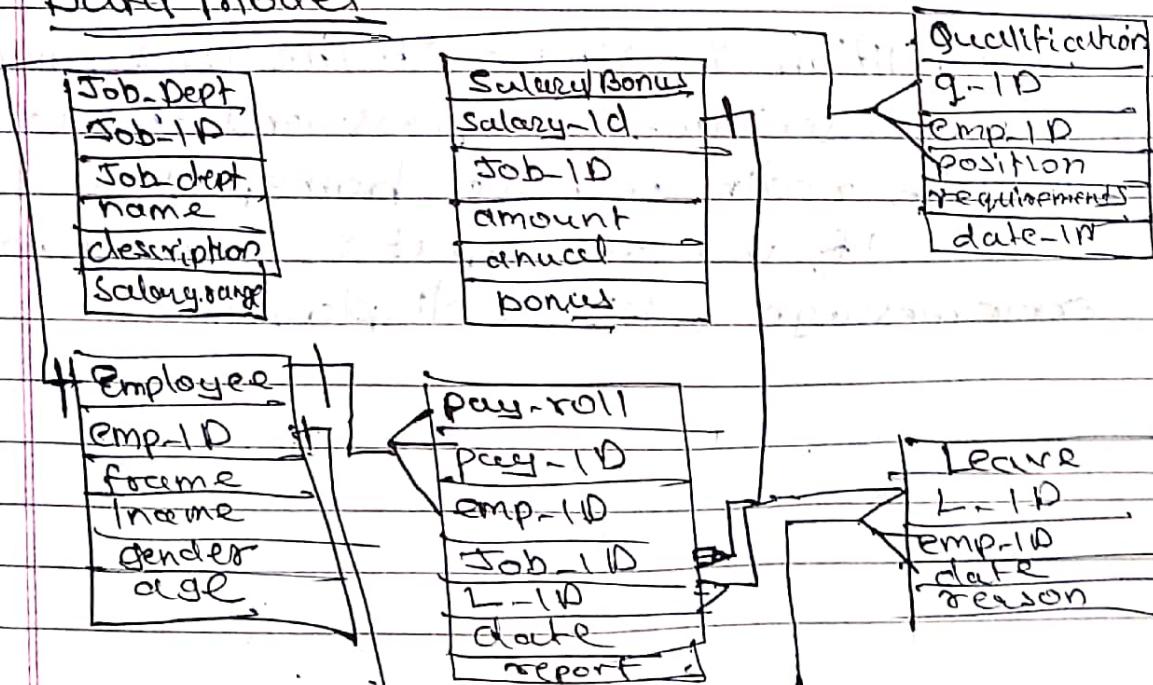
Modality

modality of relationship is, if there is no explicit need of relationship to occur or relationship is optional.

- ⑥ Draw and explain conceptual data model with E-R diagram of Employee management system. (6 marks).



Data model



⑦ Explain with example Decision table (firm).
using testing

⇒ Decision table is software testing technique used to test system behaviour for different input combinations.

This is systematic approach where the different input combinations and their corresponding system behaviour (output) are captured in tabular form.

That is why it is also called as Cause-Effect table where cause and effects are captured for better test coverage.

Example 1: Decision Base Table for Login Screen.

Email

Password

The condition is simple if the user provides correct username and password the user will be redirected to homepage.

If any of input is wrong, an error message will be displayed.

Decision Table

Condition	Rule 1	Rule 2	Rule 3	Rule 4
Username (T/F)	F	T	F	T
Password (T/F)	F	F	(M/F)	F
Output (E/H)	E	E	F	H

Legend

T - Correct username / password

F - Wrong username / password.

E - Error message is displayed

H - Home screen is displayed.

Interpretation

- Case 1 : Username and password both were wrong. The user is shown an error message.
- Case 2 : Username was correct, but the password went wrong. The user is shown an error message.
- Case 3 : Username went wrong, but the password was correct. The user is shown an error message.
- Case 4 : Username and password both were correct, and the user navigated to homepage.

⑧ Explain structured flowchart with suitable example. (4m)

or

Describe notations used for preparing a structured chart. (4m)

→ Structured chart represents the hierarchical structures of module.

It break down the entire system into the lowest functional module and describes the functions and sub-functions of each module of system in greater detail.

The article focuses on discussing structure chart in details.

Structured chart partitions the system into black boxes (functionality of system is known to users, but inner details are unknown).

① Inputs are given to the block box and appropriate output are generated.

② modules at the top level are called modules at low level.

③ Components are read from top to bottom and left to right.

④ When a module calls another, it views the called module as a Black Box, passing the requirement parameters and receiving results.

Symbols/notations used in flowchart

① module

It represents the process or task of the system. It is of three types.

a) Control module

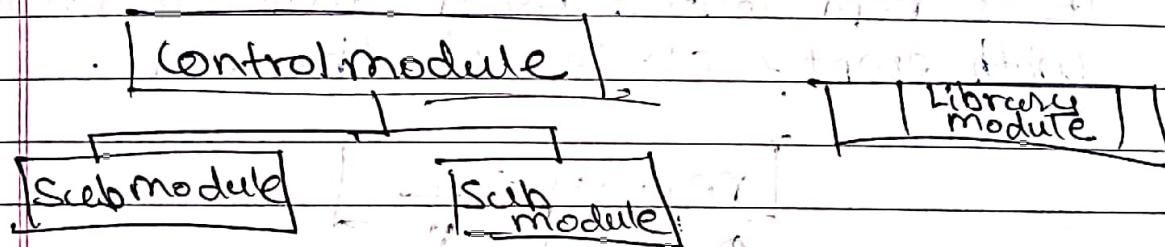
A control module branches to more than one sub module.

b) Sub module

A sub module is module which is the part(child) of another module.

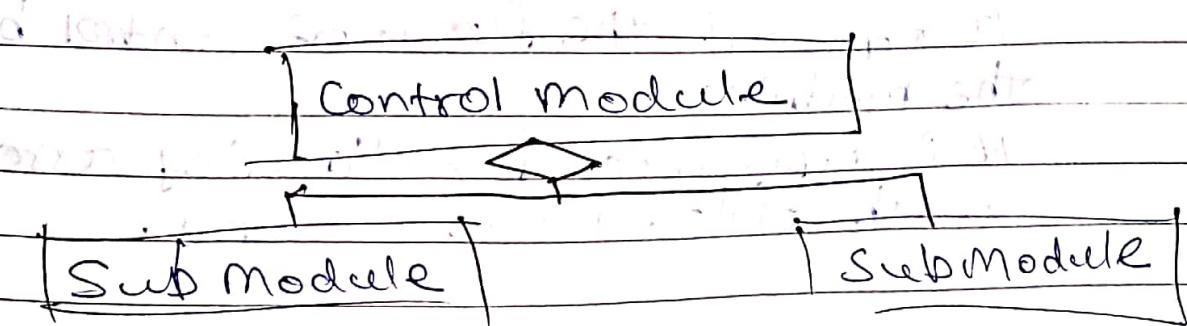
c) Library module

Library module are reusable and invokeable from any module.



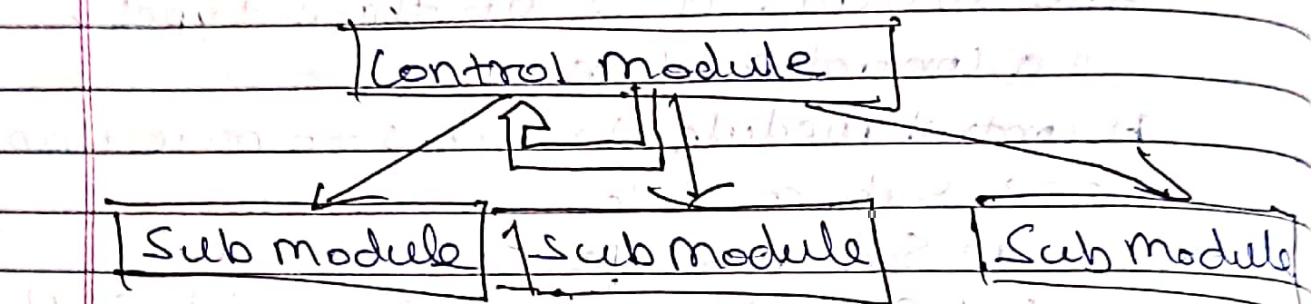
② Conditional Call

It represents that control module can select any of the sub module on the basis of some condition.



③ Loop (Repetitive Cell of module)

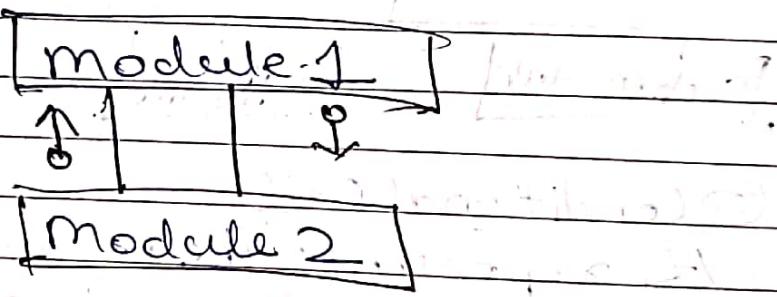
It represents the repetitive execution of module by sub module.



④ Data flow

It represents the flow of data between the modules.

It is represented by a directed arrow with empty circle at the end.



⑤ Control Flow

It represent the flow of control between the modules.

It is represented by directing arrow with filled circle at end.

Module 1

Module 2

⑥ physical storage

Physical storage is where all the information are to be stored.

Physical storage

Example, Structured chart for an Email-server;

