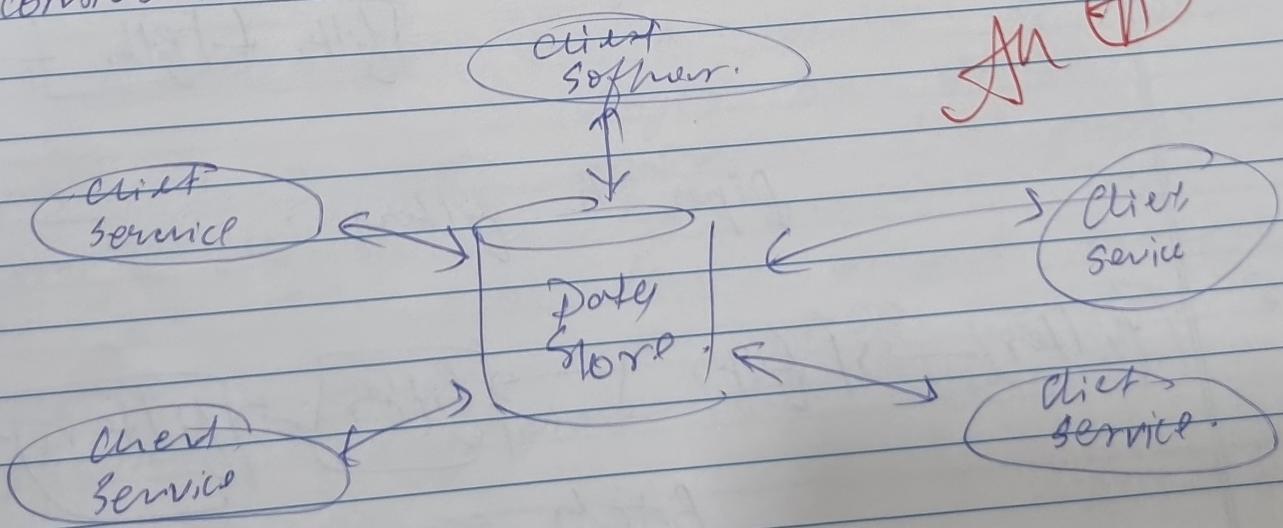


-1 Assignment :-

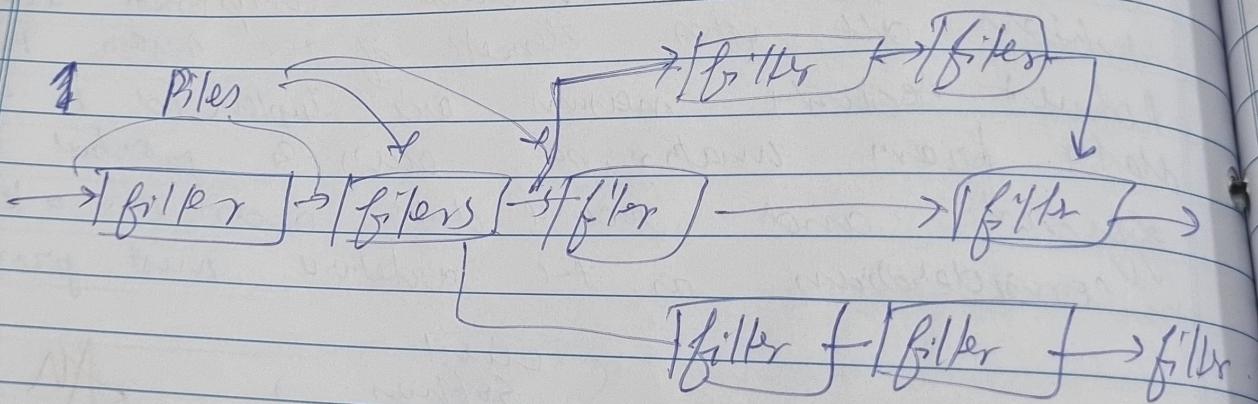
Architectural Design: Explain in details each type with example.

The s/w architecture of a program or computer system is the struct or struc. of the sy. which comprise the s/w components for externally visible properties of those components and the relationships among them.

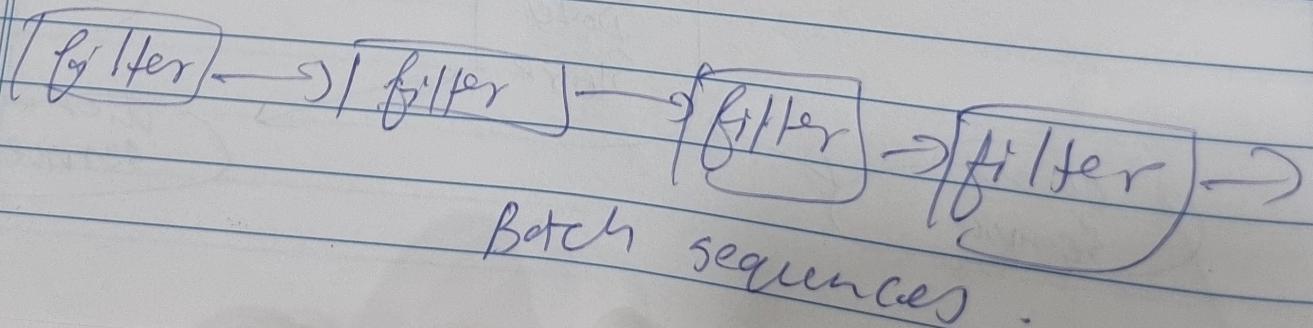
1) Data Centred architecture: A data centred architecture places data at the forefront of system design with data being stored and accessed as its imp. focus. In this architectural approach, data is considered the central and most valuable component around which all other elements of the system revolve. Robust security measures are implemented to safeguard data from unauthorized access & parallel access is a key consideration as the architecture must provide.



(2) Data flow architecture: This archi. is often used when the input data needs to be transformed through a series of computation or manipulation components into output data. It uses filter patterns as a set of components that independently handle upstream and downstream data and is designed to expect data input. The filter does not require knowledge of the working of its neighbors or of the execution context. It encourages decoupling and modularization with this design concurrent execution is supported.



Pipes & filters.

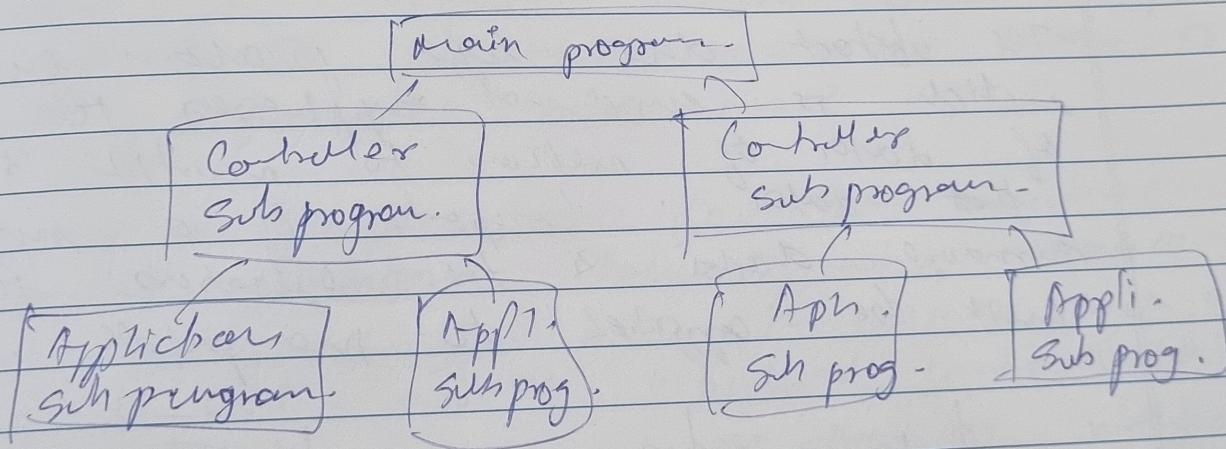


Batch sequences.

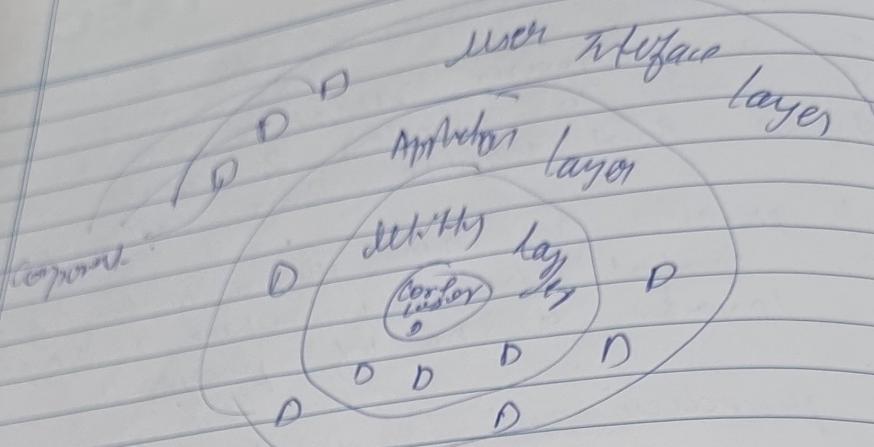
Cell & ection architecture:- Style enable a program design to achieve a program structure that is relatively easy to modify & scale two. sub-styles exist within this category.

① main/sub program architecture: program structure function into a control hierarchy where a "main" program invokes a no of programs.

② Remote procedure call architecture: The components of main program sub program architectures are distributed across.

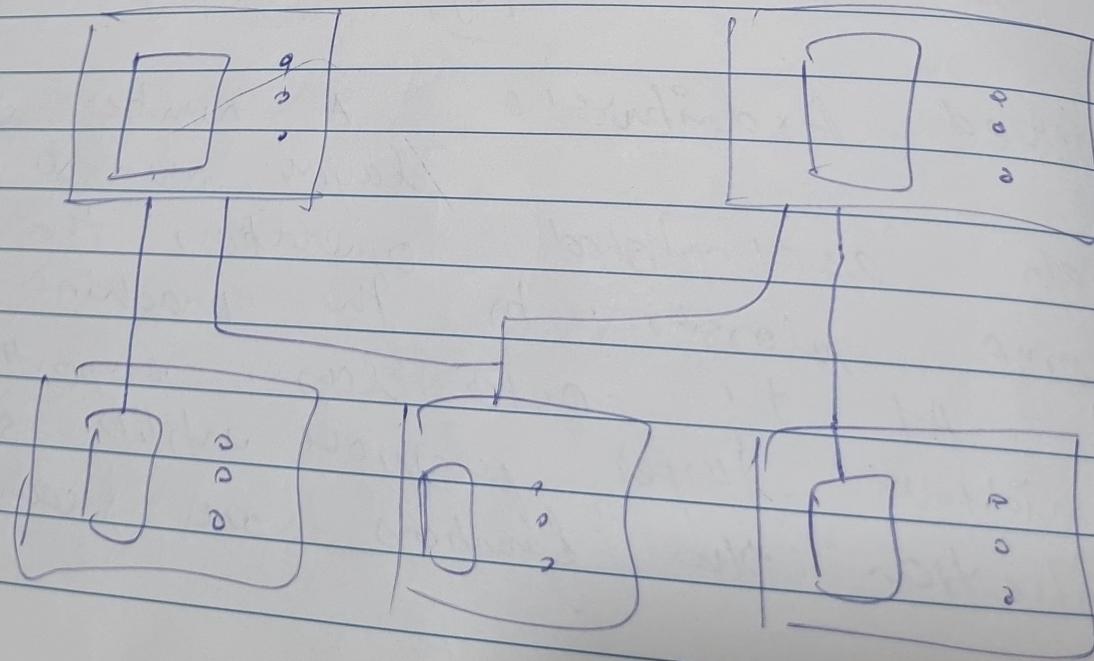


Layered Architecture: A number of diff. layers are design. operations that progressively accomplish the machine inst. become closer to the machine inst. Set. At the outer layer components. example. Intermediate layer provide utilib services & application the functions are common.



② Object oriented architecture:

The abstract data type paradigm from which is universal emphasizes the building of data of various forms to maintain & access numerous data & the operations thereon must be applied to manipulate.



Assignment 2

Ques :- Explain & detail:-

S/w maintenance :-

product after it has been delivered to the customers. It is the process of modifying a s/w product after it has been delivered to the customers.

The main purpose and update of the maintenance is to modify correct faults of applications after delivery and to improve performance.

It is done after no product has s/w related faulted for several reasons including improvement of s/w, overall correctness issues or bugs, to boost performance and more.

Following are some reasons for the maintenance -

- 1) Market Conditions: Policies, which changes over few times such as taxation and rules introduced contracts like, Reinforcement bookkeeping may trigger need for modification.
- 2) Client requirements: Over the time, customer may ask for new features.

3) Platform modifications:- Any of the platform (such as OS) or part of the system that changes are needed to keep adaptation.

4) Organisation changes: Any business level change at client side in original s/w

There are 4 types of SW maintenance
as Perpetual maintenance: A SW product needs
new features that the users want

b) Adaptive maintenance:- Involves modification
and updating
the customer need no product to run
on new platform or new OS.

c) Perfective maintenance: A system or product
needs maintenance to
support the new features.
Want or to change diff. type of system

d) Preventive maintenance:- This maintenance includes
modification and update
to prevent future problem of SW. It
goes to afford problems which are not
seen at the moment

* Re-Engineering :- When we need to
update the SW to
fit to current market without impacting it
functionality if called SW re-engineering

It is a thought process where the design
of SW is changed and program is

Ex. initially this was delivery in assembly long. when long c. came into because working in assembly was difficult.

Other than this, sometimes programmers notice that few parts of SW need more maintenance than others. Any they also need de-engineering.

The 5 steps are:

- i) Decide what to re-engineer. It is whole SW or a part of it.
- ii) Perform Reverse Engg. in order to obtain specification of existing SW.
- iii) Restore Program if required.
- iv) Re-structure codes as required.
- v) Apply forward engineering concept in order to get re-engineering done.

Reverse Engg. :- It is a process to achieve by spec. through analyzing, understanding the early sys. This process can be seen at three SDLC model. i.e. we by abstraction levels.

SW Reverse Engg. is the process of recovering the design and the requirements specification of a project from an analysis of its code.

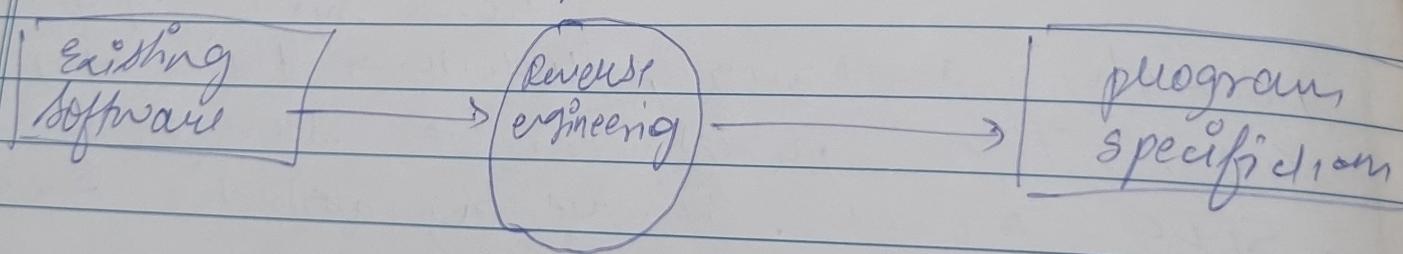
The existing system is previously implemented
 design agent which we have from hotfix
 get the design with design in hand
 they try to code to system specifications
 Some reasons for reverse engg.

- Providing proper system documentation
- Recovery of lost info.
- Assist in maintenance.
- Failure of software license.
- Discovering unexpected flow or faults.

S/w reverse engg is the process of recovering
 the design by the requirement of specification
 a product from an analysis of its code.

Reverse engineering is usually done on previously
 implemented design engg. by looking at
 the code and try to get the design.

Diagram for R/E process



Reverse Engineering of S/W is a tedious part
 because usually the S/W is funded via
 obfuscation.