

## Assignment NO: 02

Q] What are the objective of Software Engineering?

ANS:-

- i] Software design is a mechanism to transfer user requirement into some suitable form which helps the programmer in software coding and implementation.
- ii] It deals with the client requirement into SRS document. i.e. easily implemented using programming language.
- iii] Software design is first step in SDLC.
- iv] In Software design, we consider the system to be set of components or modules with clearly behaviour & boundary.

Correctness

Completeness

Efficiency

Flexibility

Consistency

Maintainability

fig:- objective of the Software Design.

1] Correctness:-

Software design should be correct as per user requirement.

2) Completeness:- The design should have all kind of components like data structure, model and external interface.

- 3] Efficiency :- Resource should be used efficiently by the program
- 4] Flexibility :- Able to modify on changing needs.
- 5] consistency :- There should be no inconsistency in the design.
- 6] Maintainability :- The design should be so simple so that it can be easily maintainable by other designer.

Q2] What are the quality attribute of software designing concept?

ANS:-

The quality attribute of design name as "PSURF" are as follows:-

- 1] Performance :- It is measured by considering process speed, response time, resource consumption, throughput and efficiency.
- 2] Supportability :- It combines the ability to extend the program, adaptability, serviceability. These three terms defines the maintainability.
- 3] Usability :- It is access by considering the factors such as human factor, overall aesthetics, consistency and documentation.
- 4] Reliability :- It is evaluated by measuring parameters like frequency and security of failure, output result accuracy, the mean-time-to-failure (MTTF), recovery from

Q3]

Ans

failure and the program predictability.

5] Functionality :- It evaluates feature set and capabilities of the program.

Q3] Explain parameters of Software designing concept?

Ans:-

- i) Concept are designed as a principle idea or invention that comes into our mind or in thought to understand something.
- ii) Idea or principal behind the design.
- iii) There are many concept of software design and some of them are given below:-

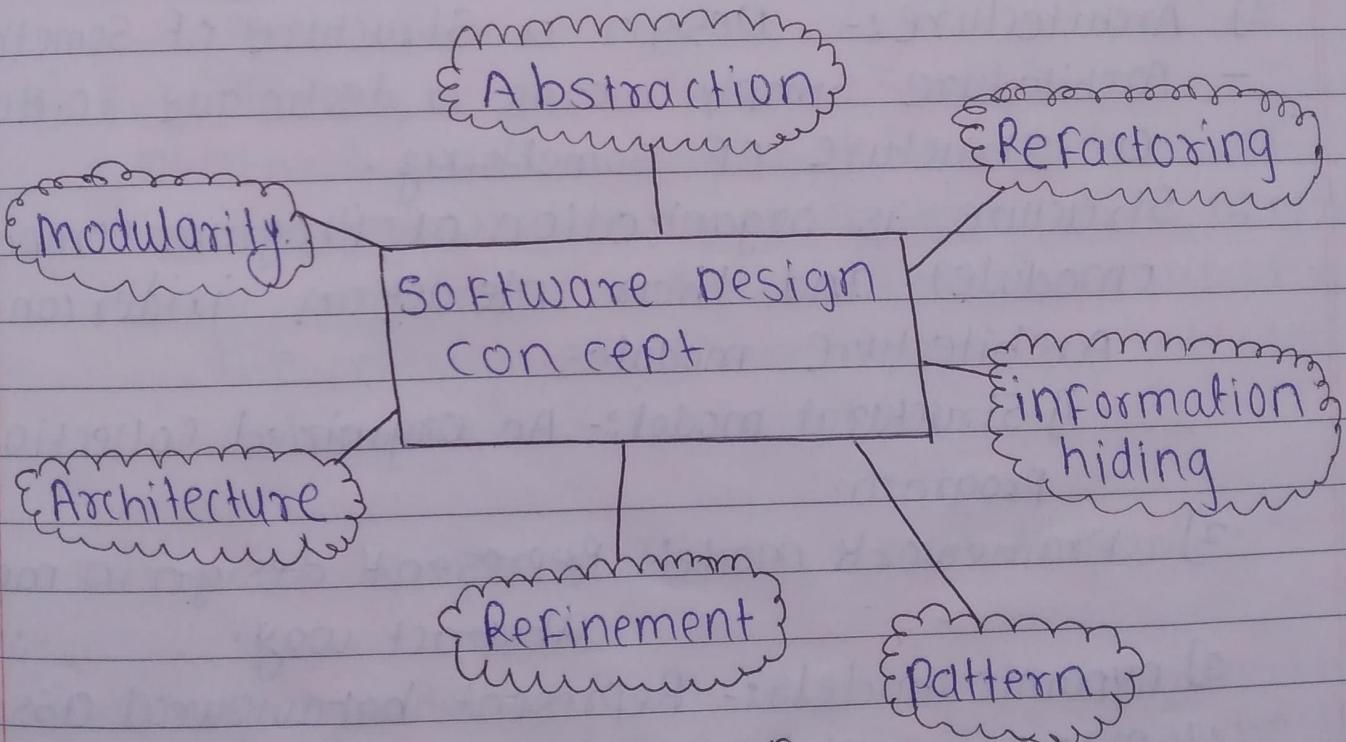


Fig:- Software Designing Concept

1] Abstraction :- It is a process of hiding complex properties of the characteristics of the software itself to keep thing more simplistic.

- The developer will be able to hide complicated and unnecessary details in the background while retaining core information in the foreground.
- Abstraction simply means to hide the details to reduce complexity and increases efficiency or details.

2] Modularity:- modularity means divide the system or project into smaller parts to reduce the complexity of the system or project.

- means Subdividing a system into smaller parts and use them independently in different system to perform different function.

3] Architecture:- Design a Structure of Something.

- Architecture Simply means a technique to design a structure of something.

- Structure is organization of program components (module) and their interaction interconnection architecture modules.

i] Structural model:- An organized collection of program.

ii] Framework model: Represent design in more abstract way.

iii] Dynamic models:- Represent behavioural aspect.

iv] Process model:- focus on the design of the business or technical process.

v] Refinement:- Removes impurities.

i] Refinement Simply means to refine something to remove any impurities if present and increase the quality.

- Refinement is very necessary to find out any error if present and then reduce it.

5] pattern :- A repeated form.

The pattern in the design process means the replication of a solution to a common recurring problems within a certain context.

6] Information hiding :- Hide the information.

Information hiding simply means to hide the information so that it cannot be accessed by an unwanted party.

- Hiding contains in such way that the information gathered or contained in one module is hidden and can't be access by any other modules.

7] Refactoring :- Reconstruct Something.

- Defined refactoring as "the process of changing a software system in a way that it won't affect the behaviour of the design and improves the internal structure."

Q4] Explain design modelling with diagram.

Ans:-

1] Design modelling in a Software Engineering represent the features of the software that helps engineers to develop it effectively, the architecture, user interface and component level detail.

- ii] modelling provides variety of different views
- iii] different method like data-driven, pattern-driven, object-oriented method are use for constructing a models.
- iv] These all methods are use to set of design principles of designing a model.
- v] Designing model is an important phase and multi-process that represent the data-structure, program structure, interface characteristic and procedural detail.
- vi]

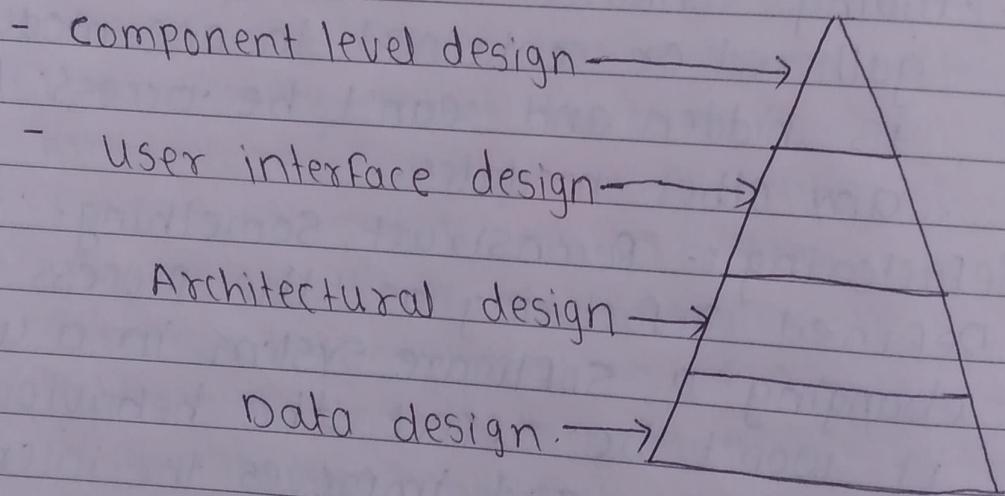
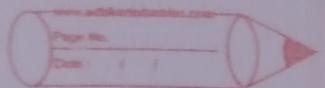


Fig:- categories of Design modelling.

Design modelling can be classified into four categories:-

- 1] Data design
- 2] Architectural design
- 3] Interface design
- 4] component-level design
- 5] Deployment-level design elements.



### 1] Data Design:-

- The design elements produced a model of data that represent a high level of abstraction.
- This model is then more refined into more implementation specific representation which is processed by the computer based system.
- The structure of the data is the most important part of the software design.

### 2] Architectural Design:-

- It represent the relationship between major structural elements of the software.
- It is about decomposing the system into interacting component.

### 3] User Interface Design:-

- It represent how the software communicate with the users. i.e. the behaviour of the system.

### 4] Component Level Design:-

- It transform the structural elements of the software architecture into a procedural description.
- It is a perfect way to share large amount of data.

### 5] Deployment Level Design:-

It shows the functionality of software and Subsystem that allocated in the physical computing environment which support the software.

Q5]

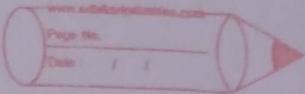
What are designing principles of software.

Ans:-

- i] Design process is a sequence of steps that enable the designer to describe all aspects of the software to be built.
- ii] The design model is created for software provides a variety of different views of computer softwares.

Design design principles:-

- i] A design should not suffer from "funnel vision". A good programmer or designer should consider alternative approaches.
- ii] The design should be traceable to the analysis model. Because a single element of the design model often traces to multiple requirements, it is necessary to have a means for tracking how requirement have been satisfied by design model.
- iii] The design should not reinvent the wheel. Design time should be invested in representing truly new ideas and integrating those patterns that already exist in the real world.
- iv] The design should exhibit uniformity and integration. Rules of style and format should be defined for a design team before design work begins.
- v] The design should be invested in representing truly new ideas and integrating those patterns that already exist.



- vi] The design should "minimize the intellectual distance" between the software and the problem as it exists in the real world.
  - vii] The design should exhibit uniformity and integration. Rules of style and format should be defined for a design team before design work begins.
  - viii] The design should be structured to accommodate change.
  - ix] The design should be structured to degrade gently even when aberrant data, events or operating conditions are encountered.
  - x] Design is not coding and coding is not design. The only decisions made at the coding level address the small implementation details that enables the procedural and design to be coded.
- Q6] Explain taxonomy of software architecture with diagram.
- Ans:-
- i] This design has been described as a multi-step process in which representation of data and program structure, interface characteristics and procedural detail are synthesized from information requirement.
  - ii] There are many architectural styles, each style will describe a system category that consist of.

i] A set of components (e.g. database computational modules) that will perform a function required by the system.

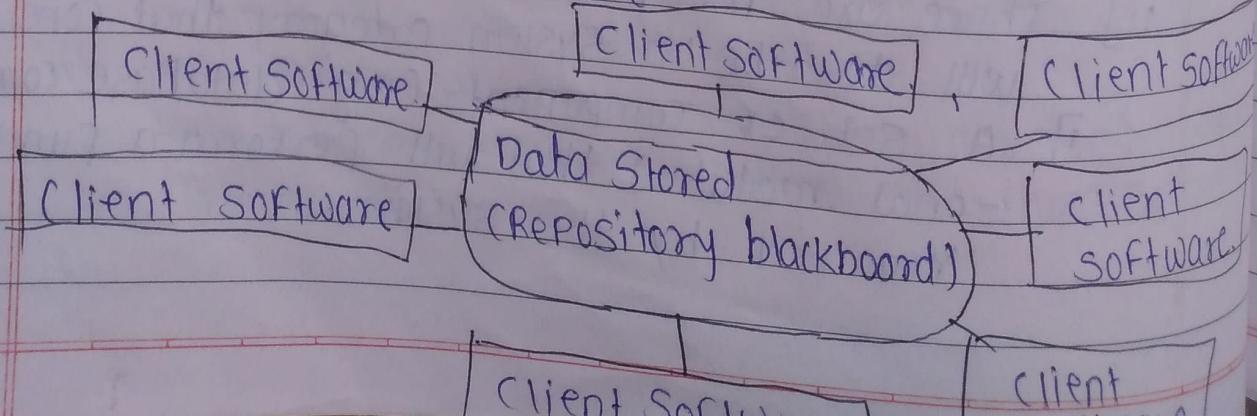
- 2] The set of connection will help in coordination, communication, and cooperation between the component.
- 3] conditions that how components can be integrated to form the style.
- 4] semantic models that help the designer to understand the overall properties of the system

### Taxonomy of Architectural Styles and Patterns:-

- 1] Data centered architectures.
- 2] Data flow architectures.
- 3] call and return architectures
- 4] object oriented architectures
- 5] Layered architectures.

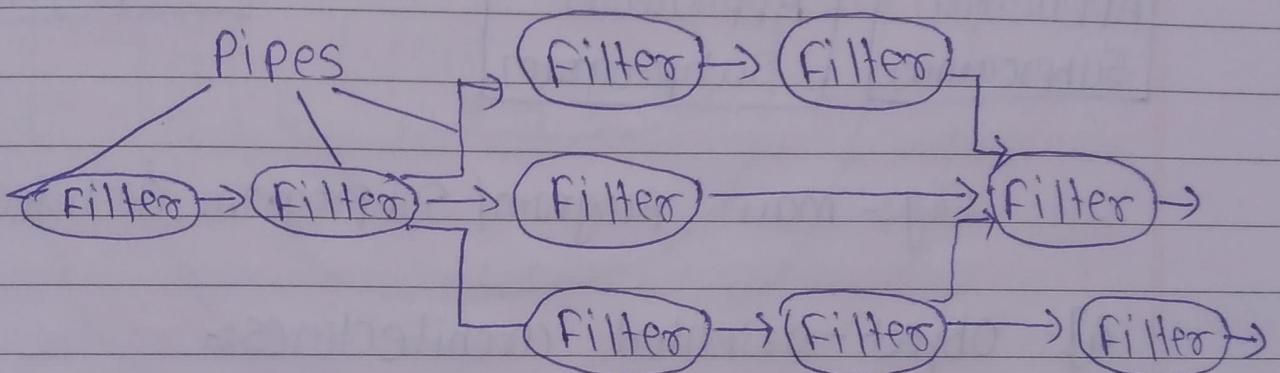
#### 1] Data centered architectures:-

- 1] A data Store (e.g. a file or database) resides at the center of this architecture and is accessed frequently by other components that update, add, delete or otherwise modify data within the store.
- 2] Client Software access the data independent of any changes to the data or the actions of other client software.

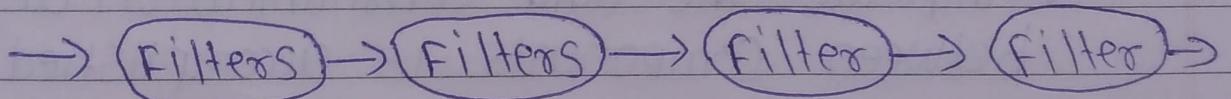


## 2] Data Flow architecture:-

- 1) This architecture is applied when input data are to be transformed through a series of computational or main manipulative components into output data.
- 2). Each component known as filter, transform the data sends this transformed data to other filters for further processing using the connector known as pipe.



(a) Pipes and Filters.



(b) Batch Sequential.

## 3] call and return architecture:-

A call and return architecture enables software designers to achieve a program structure, which can be easily modified. This style consists of the following two sub styles.

- a) Main program / Subprogram architecture.
- b) Remote procedure call architecture.

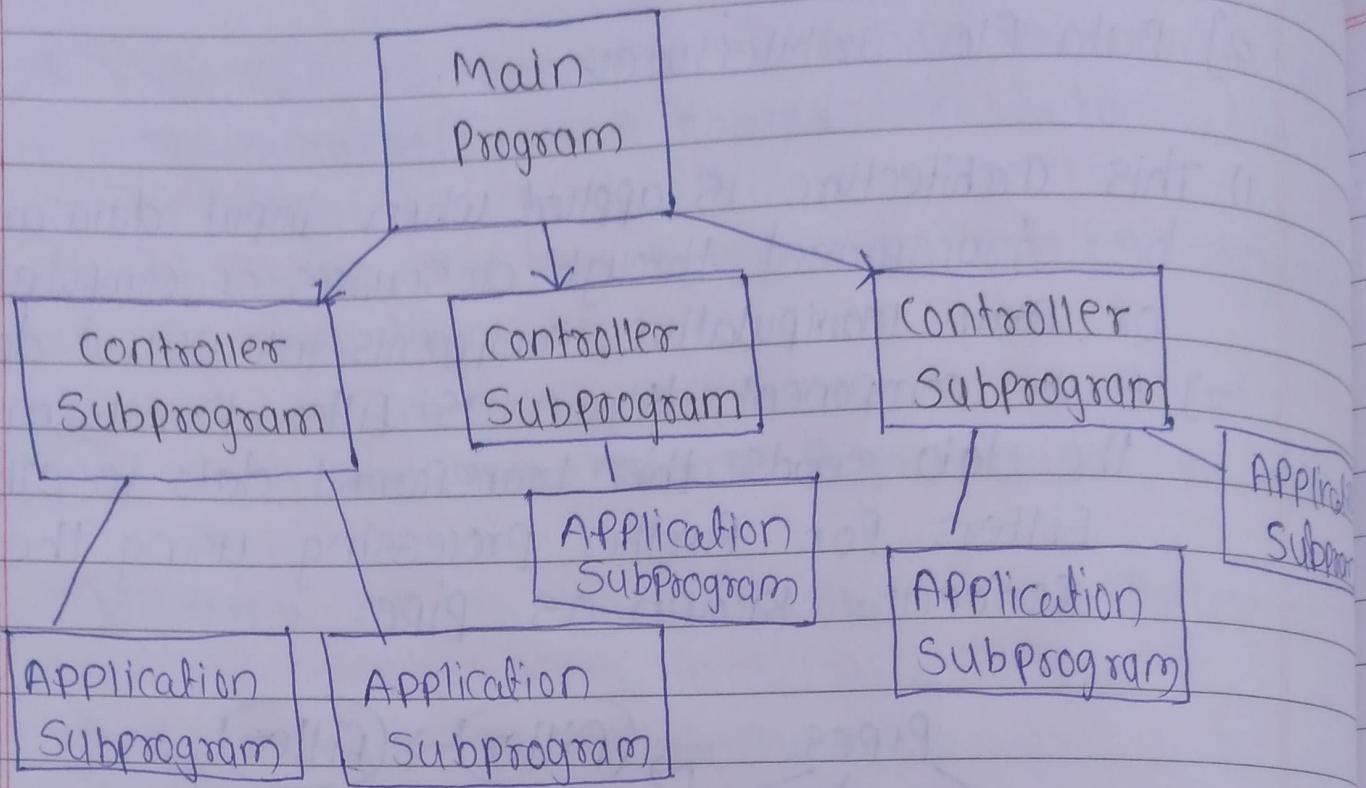


Fig:- main program/ subprogram architecture.

#### 4] Object - oriented architectures:-

The Components of a System encapsulate data and the operation that must be applied to manipulate the data. Communication and coordination between components is accomplished via message passing. Object- protect system integrity.

#### 5] Layered Architecture:-

- i) In Layered architecture, Several layers (components) are defined with each layer performing a well defined set of operation.
- ii) Each layer will do some operations that becomes closer to machine instruction set progressively.

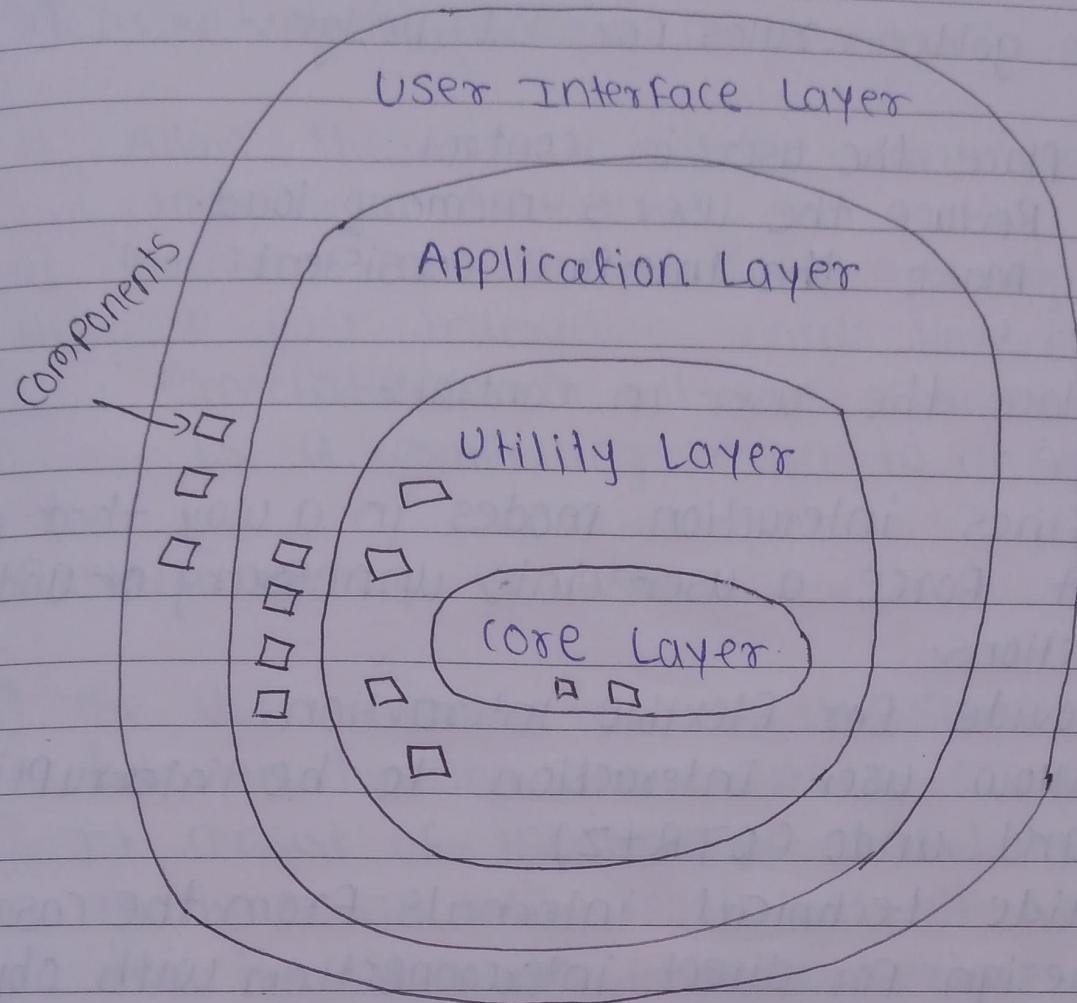


Fig: Layered Architecture.

Q7] What is UI design? Explain 3 golden rule.

Ans:-

i] User interface (UI) design is the process designer use to build interfaces in software or computerized devices, focusing on looks or style.

ii] Designer aim to create interfaces which users find easy to use and pleasurable.

## \* The golden rules for UI Design:-

- 1] Place the user in control.
- 2] Reduce the user's memory load.
- 3] Make the interface consistent.

### 1] place the user in control:-

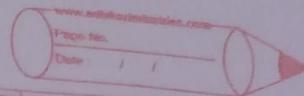
- i) Defines interaction modes in a way that does not force a user into unnecessary or undesired actions.
- ii) provide for flexible interaction.
- iii) Allow user interaction to be interruptible and undo (CTR+z)
- iv) Hide technical internals from the casual user
- v) design for direct interconnection with objects that appear on the screen.

### 2] Reduce the user's memory Load:-

The more a user has to remember, the more error prone will be interaction with the system.

- Principles that enables an interaction interface to reduce the user's memory load:-

- 1) Reduce demand on Short-term memory.
- 2) Establish meaningful defaults.
- 3) Define shortcuts that are intuitive.
- 4) The visual layout of the interface should be based on a real world metaphor.



3] Make the interface consistent:-

- i] Allow the user to put the current task into a meaningful context.
- ii] Maintain consistency across a family of application.
- iii] If past interactive models have created user expectations, do not make changes unless there is a compelling reason to do so.

8] Explain various UI users:-

Ans:-

- i] The User interface (UI) is the space where interaction between humans and computers occurs.
- ii] It consists of information output from the machine as well as a set of control elements for the user to perform certain actions.

Types of UI users:-

- i] User interface of command line interface (CLI):-  
This type of user interfaces allows the user to interact directly with the computer by typing commands. Typing command is not that easy because just can't type anything, you have to type very specific words, so that computer is able to understand.

Advantages:-

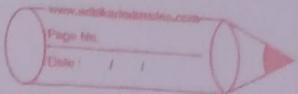
- 1] It is faster than the other types of UI.
- 2] Lesser memory (RAM) is used.
- 3] It doesn't need windows to run.
- 4] Lesser CPU processing time is needed.

## 2] Graphic user interface :-

- i] A Graphical user interface (GUI) is the most common type of user interface in use today.
- ii] It is a very 'friendly' way for people to interact with the computer because it makes use of pictures, graphics and icons - hence it is called graphical.
- iii] This type of UI is easy to use specially for a beginner.
- iv] we do not have to learn complicated commands.

## 3] Menu-driven interfaces:-

- i] This type of interface lets us interact with a computer or device by working your way through a series of screens or menus.
- ii] Think about our iPod, mobile phone they both use a menu driven interface.
- iii] Menu driven interface can also be verbal rather than visual.
- iv] Most of the software uses has menu interface.
- v] In menu driven there is no commands to learn.
- vi] Step by step options are given so that the user doesn't have to remember anything.



i) What is UML?

Ans:- i) The UML is a general purpose visual modelling language that is used to specify, visualize, construct, and document the artifacts of software system.

ii) It is used to understand, design, browse, configure, maintain and control information about such system.

iii) UML includes semantic concepts, notation and guidelines.

iv) UML is not a programming language, it is rather a visual language.

Q10] What are object oriented concepts in UML? in brief?

Ans:-

1] Class:- A class defines the blueprint i.e. structure and functions of an object.

2] Objects:- Objects help us to decompose large system and helps us to modularize our system.

3] Inheritance:- Inheritance is a mechanism by which child classes inherit the properties of their parent classes.

4] Abstraction:- Mechanism by which implementation details are hidden from user.

5] Encapsulation:- Binding data together and protecting it from the <sup>outer</sup> user world is referred to as encapsulation.

6] Polymorphism:- Mechanism by which functions or entities are able to exist in different forms.

Q1] What is Rumbaugh method? with class diagram.

Ans:-

- i) The Rumbaugh methodology also known as OMT (Object modelling Technique).
- ii) OMT is an approach to develop manageable Object-oriented systems and host Object-oriented Programming.
- iii) The purpose is to allow for class attribute methods, inheritance, and association to be easily expressed.
- iv) Rumbaugh / OMT consist of four stages:-
  - 1) Analysis
  - 2) System design
  - 3) Object design.
  - 4) Implementation

v) Additionally OMT is also broken down into three separate parts. These parts are:-

- 1) A dynamic model:- Object model & data dictionary.
  - 2) A dynamic model:- State diagram & event flow diagram.
  - 3) A functional model:- data flow & constraints.
- 1] An object model:- object model and dictionary.  
- Classes:- A set of individual objects.  
- Association lines:- relationship among classes.  
Class interconnected by association lines.

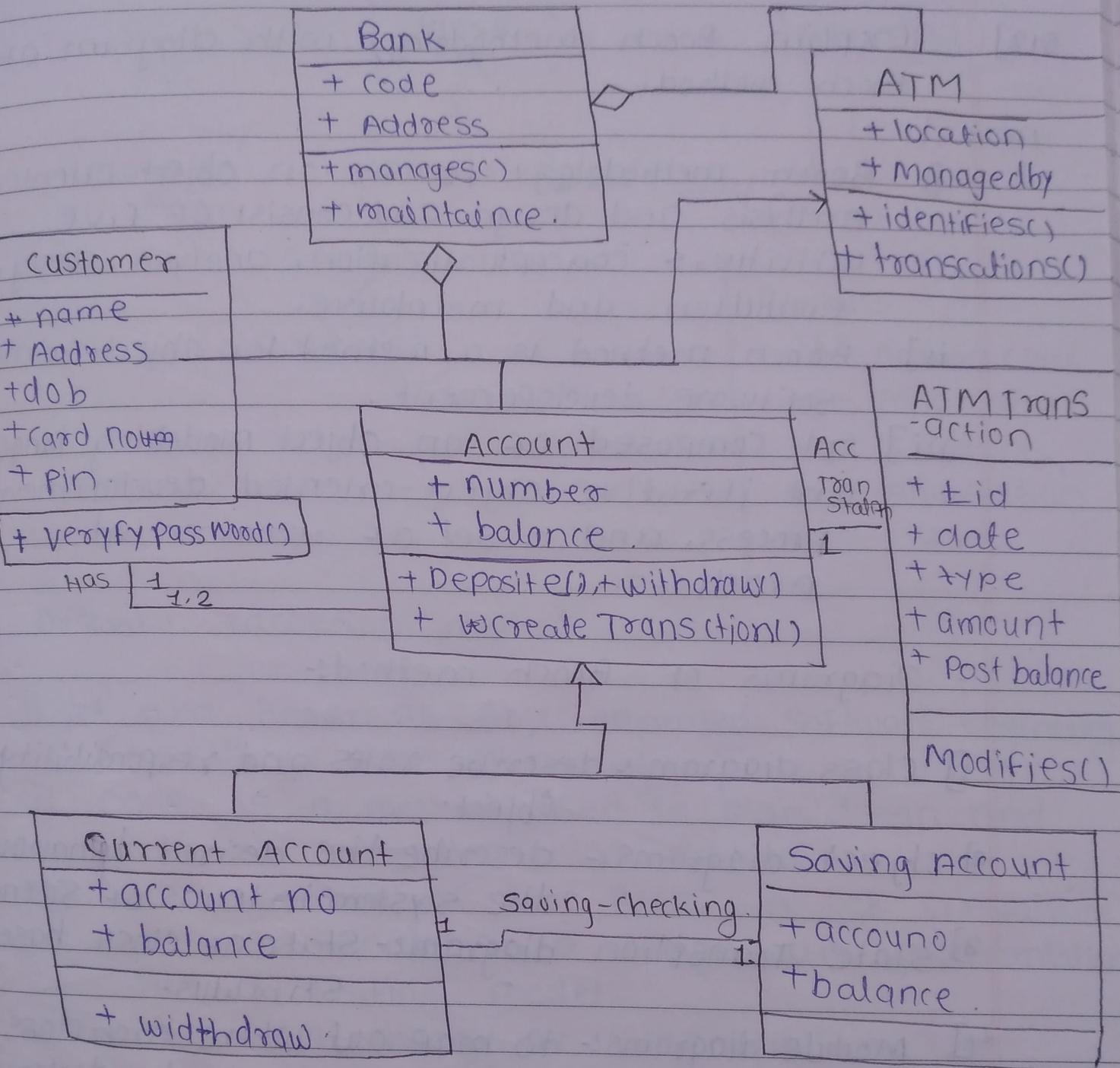


Fig: Class diagram of Bank System.

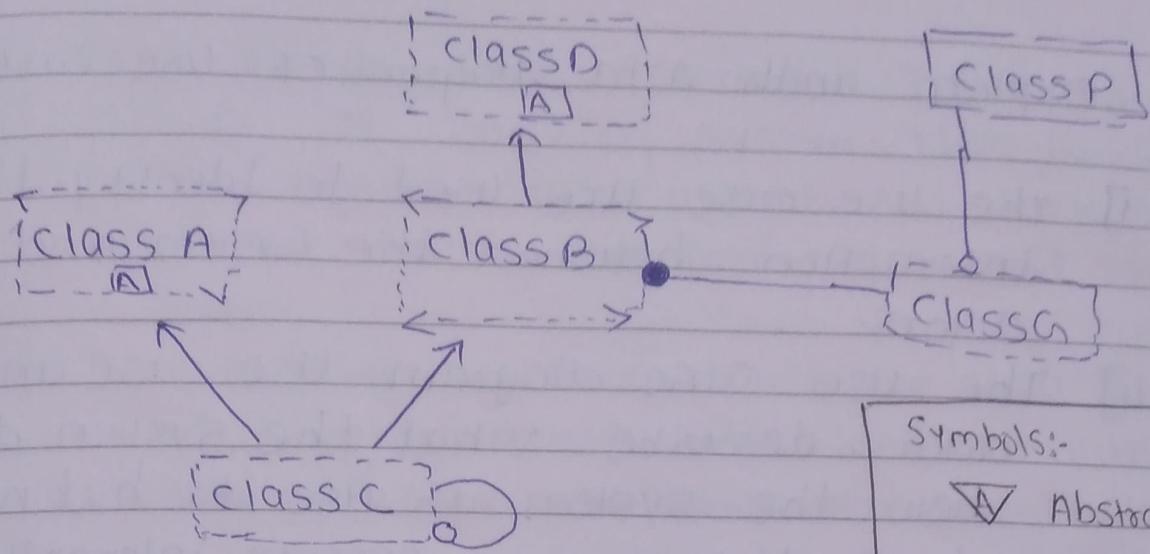
Q12] Explain Booch methodology with diagram of Booch method.

Ans:-

- i] Booch methodology focuses on object-oriented analysis and design and consist of five activities:- conceptualization, analysis, design evolution and maintenance.
- ii] Booch method is a method for object-oriented software development.
- iii] It composed of an object modelling language an iterative object-oriented development process, and a set of recommended practices.

- Diagrams of Booch method:-

- 1] Class diagram:- describe roles and responsibility of Object.
- 2] Object diagrams:- describe the desired behaviour of the system in terms of scenario.
- 3] State Transition diagram:- State of class based on stimulus.
- 4] Module diagrams:- to map out where each class & and object should be declared.
- 5] Process diagrams:- to determine to which processor to allocate a process.
- 6] Interaction diagrams:- describes behaviour of the system in terms of scenarios.



Symbols:-

- Δ Abstract class
- Association
- Inherits
- Aggregation
- Uses

Q13] Describe Jacobson method.

Ans:-

- i] It also known as object-oriented software engineering (OOSE) or even objectory.
- ii] OOSE is a method used to plan, design and implement object-oriented Software.
- iii] OOSE consist of five parts : A set of requirements, an analysis, a design, an implementation and a testing model.
- iv] The methodology of OOSE utilizes use cases in its design.

\* Use case:- use case help us understand the how we want to design our system.

- use case are scenarios for helping us understand the requirement of our system.
- we can see a use case as an interaction between a user and the system.

Q14] Explain with ATM diagram of use case.

Ans:-

- i] the use case are used to identify the interaction between the system and its actors.
- ii] The use case diagram use case and actors describe what the system does and how the system is use it. but not the how the System operate internally.
- iii] It depict the high-level functionality of system and also tells how the User handle system.
- iv) It depict the external view of the system.

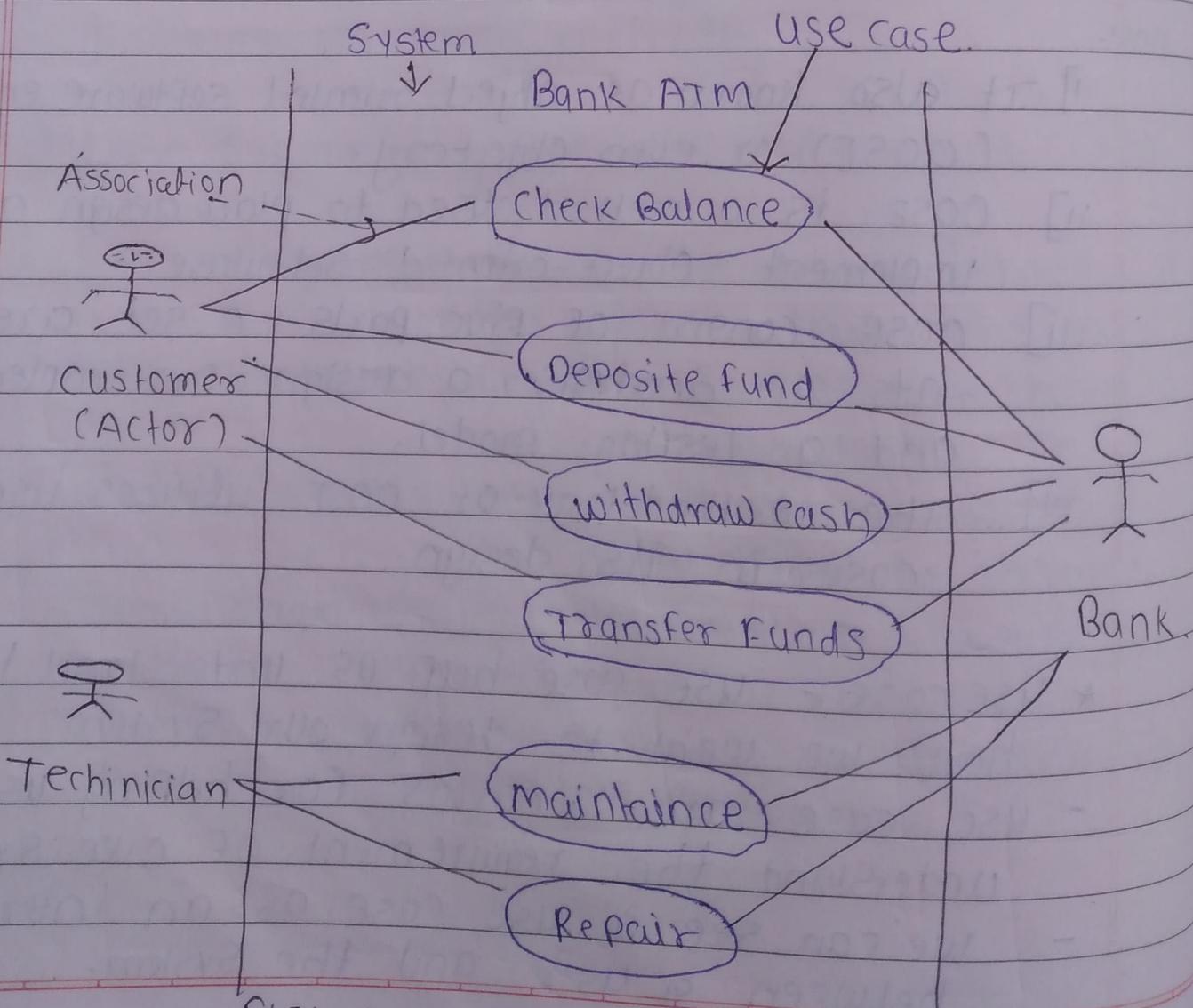


Fig:- Use case diagram

- i] ATM Stands for Automated Teller machine.
- ii] It is a banking system. This Banking System allows the customer to check Balance, Deposite Fund, withdraw cash and Transfer Fund.
- iii] In a ATM machine customer is a Actor and check Balance, Deposite Fund, withdraw cash are the use cases of the customer.

v) Describe interaction diagram along with Log-in System.

Ans:- i] Interaction diagram is use to visualize the iterative behaviour of the System.

- ii] Interaction diagram are used to establish communication between two object.
- iii] This interactive behaviour is represented in UML by two diagrams known as Sequence and collaborative diagram.
- iv] Interaction diagram is focus on a message passing.
- v] Interaction diagram capture the dynamic behaviour of the System.

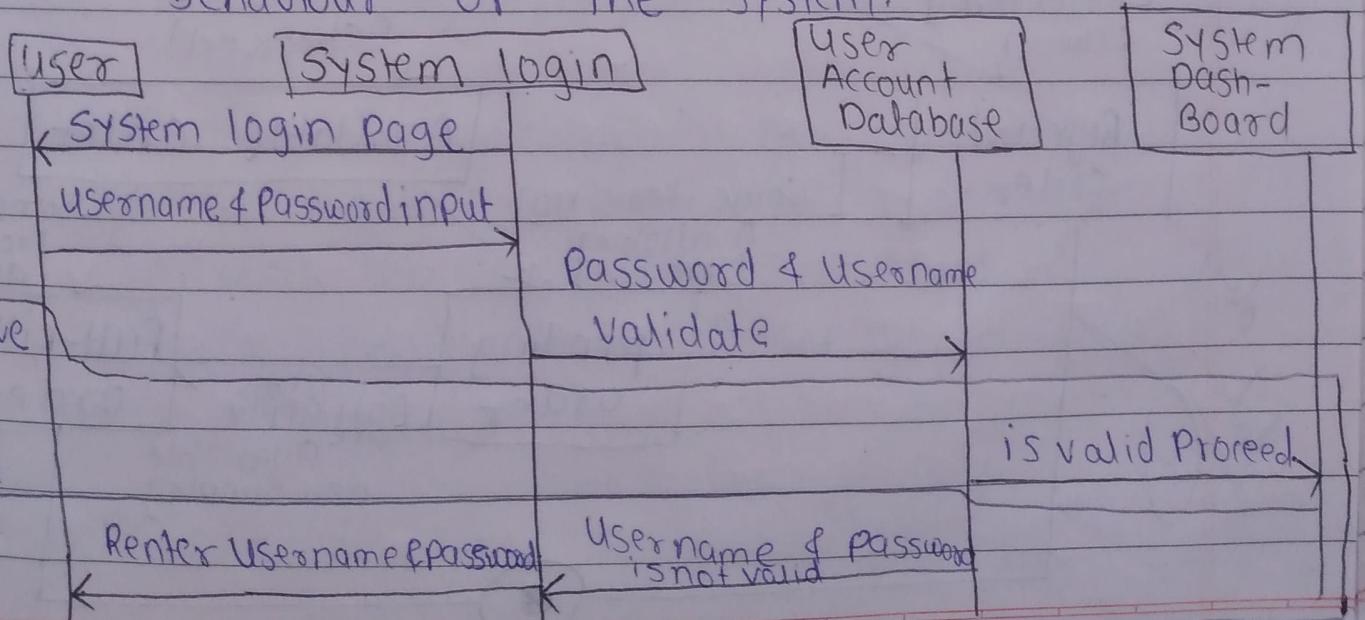


fig: Interaction diagram of log-in system.

Q16] Explain transition (changing) States of a system with any one example.

Ans:- i] This is used to represent different transition changing states of a system.

ii] It is generally used to graphically represent all possible transaction states a system can have and model such system.

iii] The system consist of various States that are being represented using various symbol in the state transition diagram

iv] Following are the symbol are use in transaction diagram:-

1] Initial State : ●

2] Final State: ○

3] Simple State: □

4] Composite State: □

