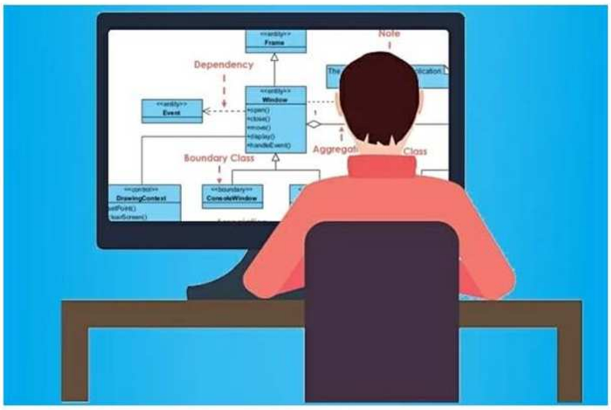
Software Design & Architecture
Spring 2022 - Week-01



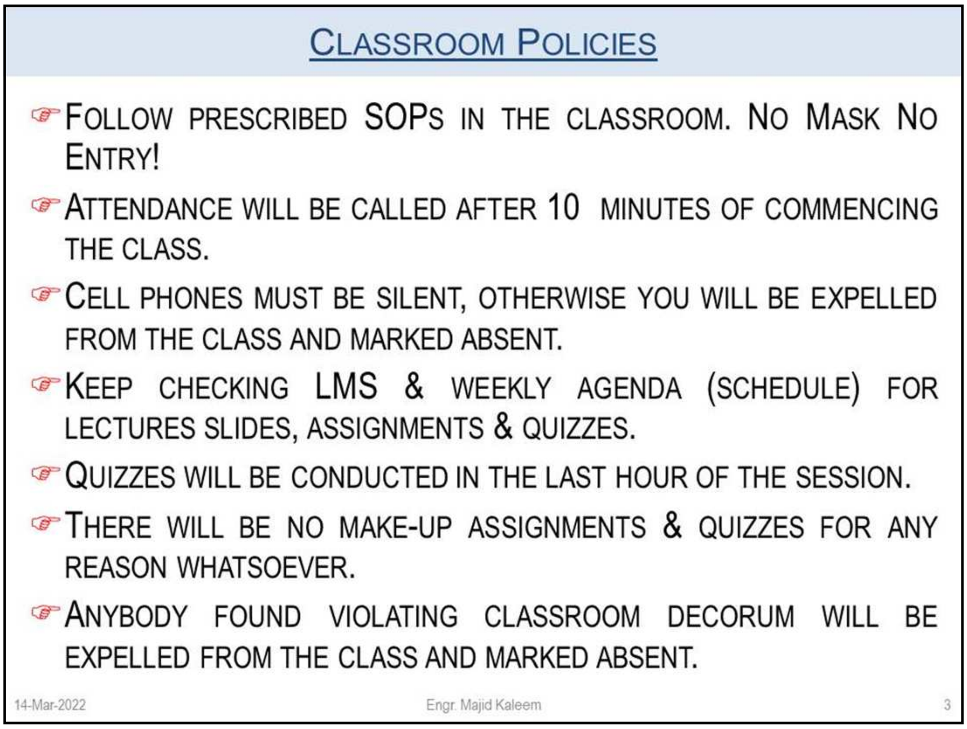
مدرس: مہندس ماجد کلیم

جامعہ بحریہ، واقعگاہ کراچی

Engr. Majid Kaleem

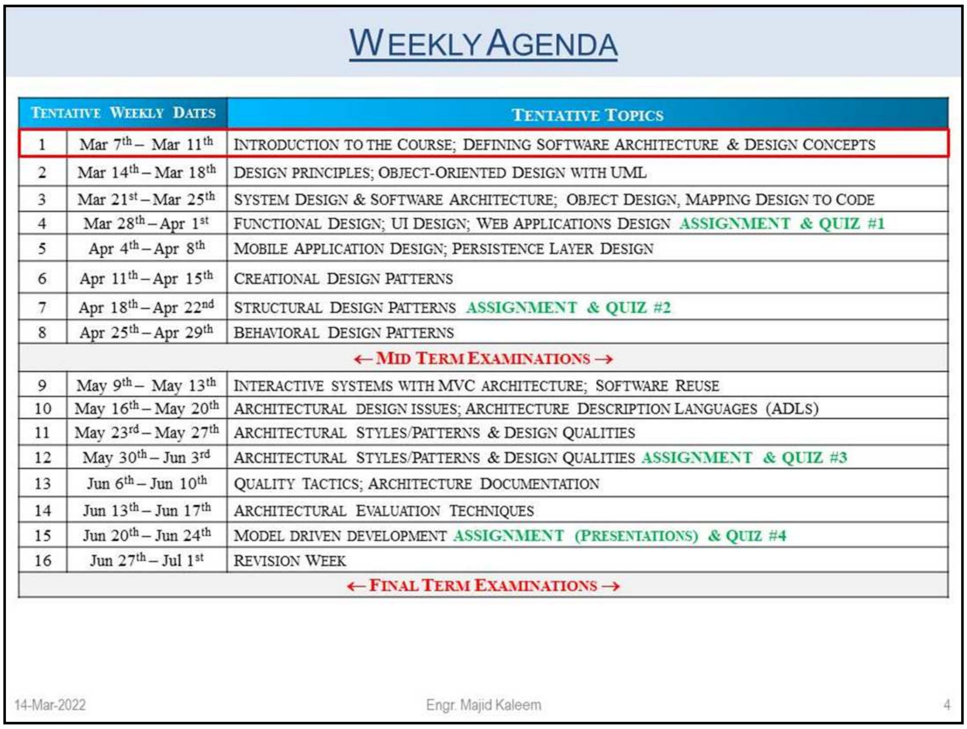
Free Slots For Counseling

Day/Time	8:30-9:30	9:30-10:30	10:30-11:30	11:30-12:30	12:30-1:30	1:30-2:30	2:30-3:30	3:30-4:30	4:30-5:30
Monday		SDA BSE-4B E-209	SDA BSE-4B E-209		SQE BSE-6A E-210				
Tuesday				SQE BSE-6A E-210	SQE BSE-6A E-210				
Wednesday		SDA BSE-4A E-204	SDA BSE-4A E-204						
Thursday									
Friday									



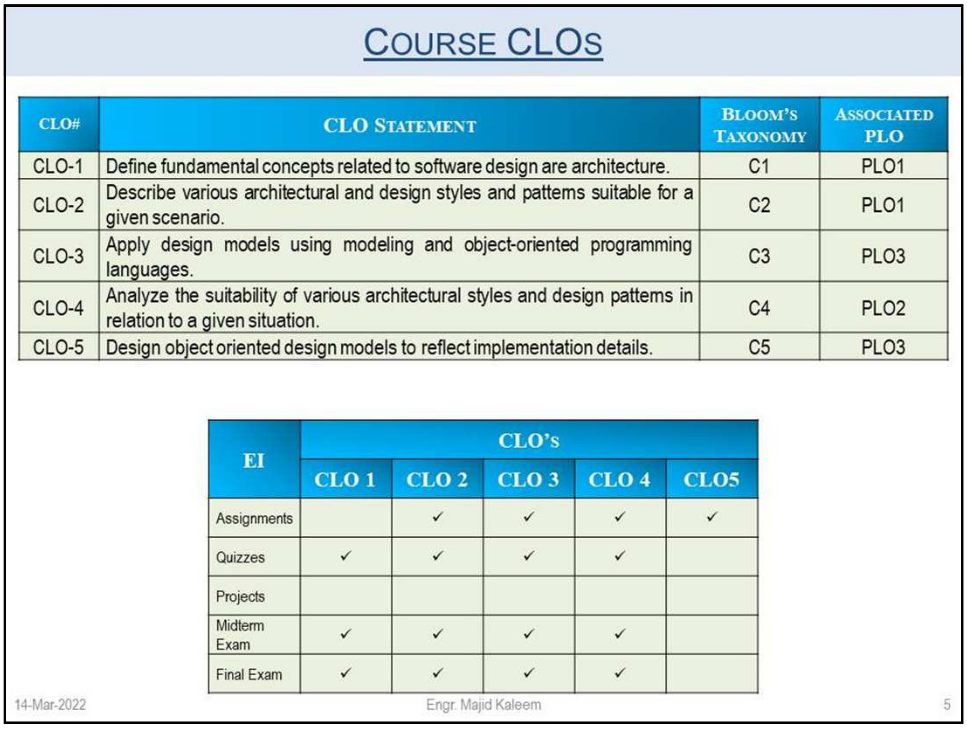
Classroom Policies

- Follow prescribed SOPs in the classroom. No Mask No Entry!
- Attendance will be called after 10 minutes of commencing the class.
- Cell phones must be silent, otherwise you will be expelled from the class and marked absent.
- Keep checking LMS & weekly agenda (schedule) for lectures slides, assignments & quizzes.
- Quizzes will be conducted in the last hour of the session.
- There will be no make-up assignments & quizzes for any reason whatsoever.
- Anybody found violating classroom decorum will be expelled from the class and marked absent.



Weekly Agenda

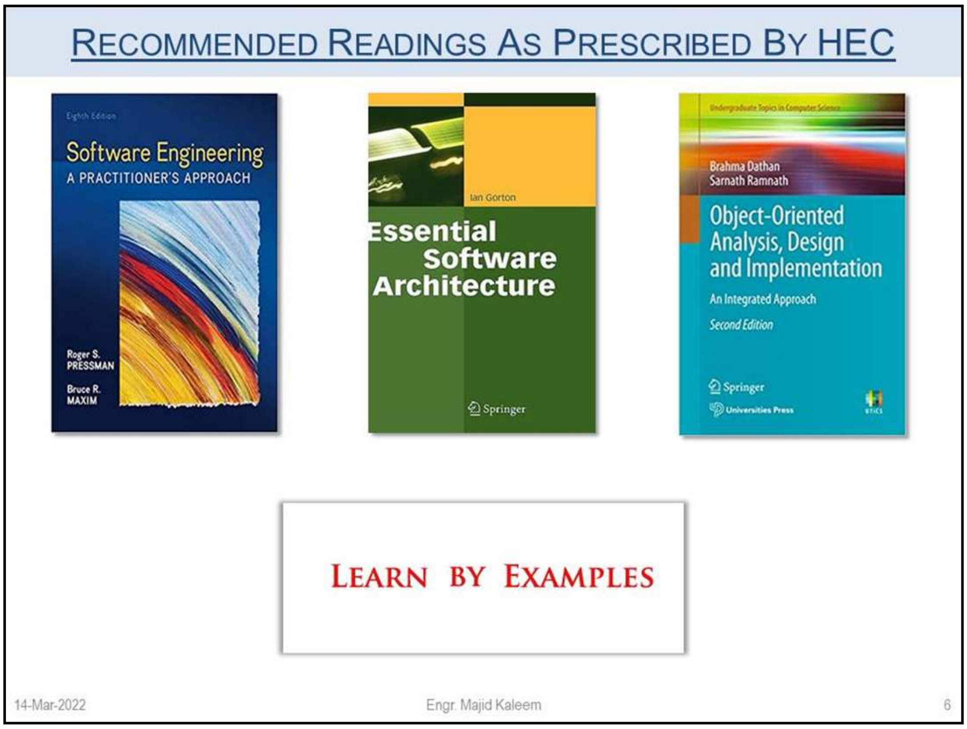
Tentative Weekly Dates		Tentative Topics
1	Mar 7th – Mar 11th	Introduction to the Course; Defining Software Architecture & Design Concepts
2	Mar 14th – Mar 18th	Design principles; Object-Oriented Design with UML
3	Mar 21st – Mar 25th	System Design & Software Architecture; Object Design, Mapping Design to Code
4	Mar 28th – Apr 1st	Functional Design; UI Design; Web Applications Design **Assignment & Quiz #1**
5	Apr 4th – Apr 8th	Mobile Application Design; Persistence Layer Design
6	Apr 11th – Apr 15th	Creational Design Patterns
7	Apr 18th – Apr 22nd	Structural Design Patterns **Assignment & Quiz #2**
8	Apr 25th – Apr 29th	Behavioral Design Patterns
		← **Mid Term Examinations** →
9	May 9th – May 13th	Interactive systems with MVC architecture; Software Reuse
10	May 16th – May 20th	Architectural design issues; Architecture Description Languages (ADLs)
11	May 23rd – May 27th	Architectural Styles/Patterns & Design Qualities
12	May 30th – Jun 3rd	Architectural Styles/Patterns & Design Qualities **Assignment & Quiz #3**
13	Jun 6th – Jun 10th	Quality Tactics; Architecture Documentation
14	Jun 13th – Jun 17th	Architectural Evaluation Techniques
15	Jun 20th – Jun 24th	Model driven development **Assignment (Presentations) & Quiz #4**
16	Jun 27th – Jul 1st	Revision Week
		← **Final Term Examinations** →



Course CLOs

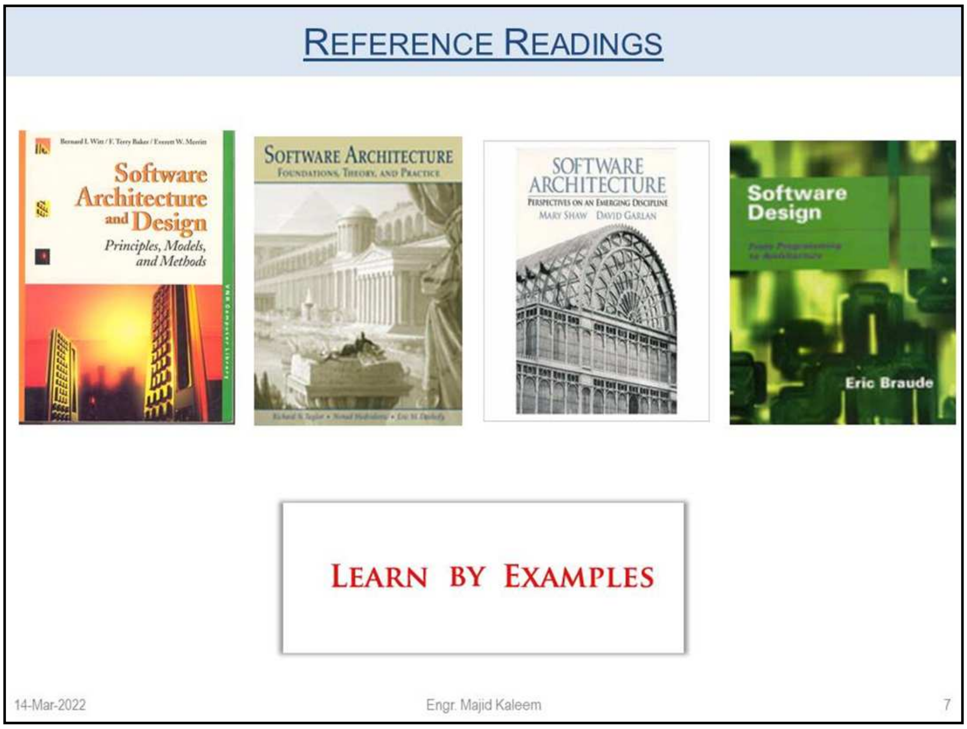
CLO#	CLO Statement	Bloom's Taxonomy	Associated PLO
CLO-1	Define fundamental concepts related to software design are architecture.	C1	PLO1
CLO-2	Describe various architectural and design styles and patterns suitable for a given scenario.	C2	PLO1
CLO-3	Apply design models using modeling and object-oriented programming languages.	C3	PLO3
CLO-4	Analyze the suitability of various architectural styles and design patterns in relation to a given situation.	C4	PLO2
CLO-5	Design object oriented design models to reflect implementation details.	C5	PLO3

EI	CLO's				
	CLO 1	CLO 2	CLO 3	CLO 4	CLO5
Assignments		✓	✓	✓	✓
Quizzes	✓	✓	✓	✓	
Projects					
Midterm Exam	✓	✓	✓	✓	
Final Exam	✓	✓	✓	✓	



Recommended Readings As Prescribed By HEC

Learn by Examples

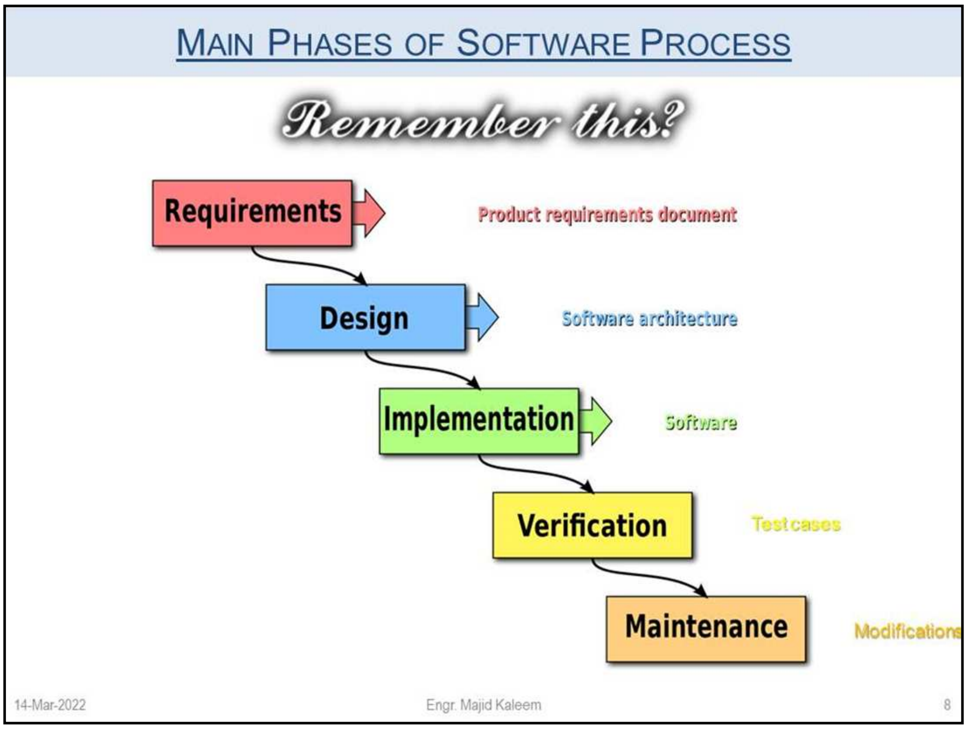


Reference Readings

Bernard I. Witt / F. Terry Baker / Everett W. Merritt
Software Architecture and Design
Principles, Models, and Methods
Software Architecture
Foundations, Theory, and Practice
Software Architecture
Perspectives on an Emerging Discipline
Mary Shaw David Garlan
Software Design
Eric Braude

Learn by Examples

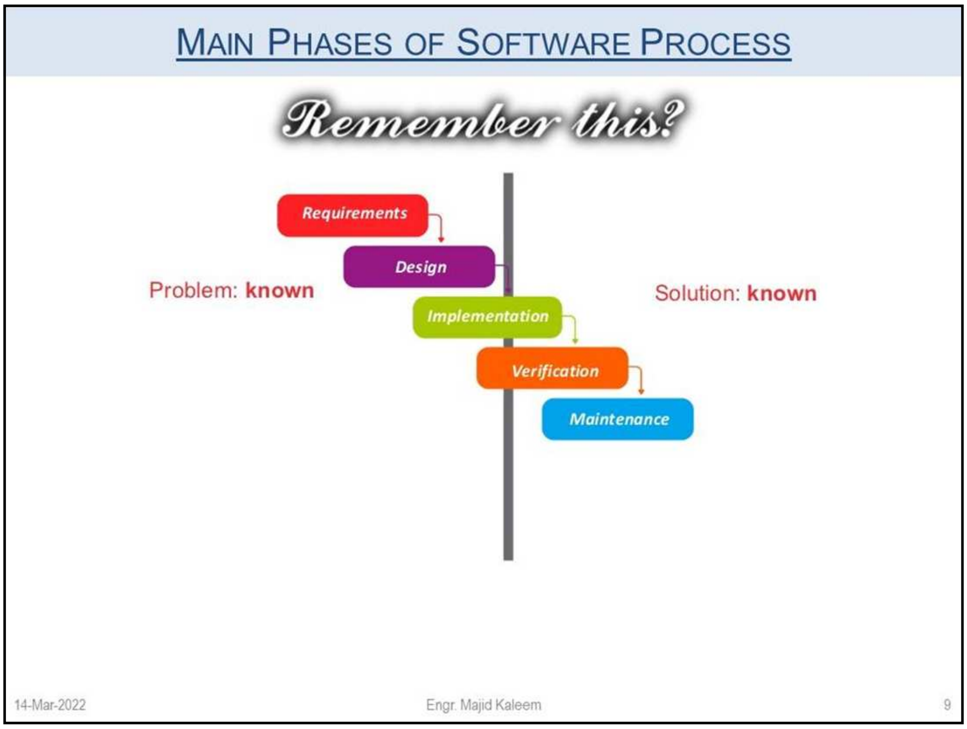
14-Mar-2022 Engr. Majid Kaleem 7



Main Phases of Software Process

Remember this?

14-Mar-2022 Engr. Majid Kaleem 8



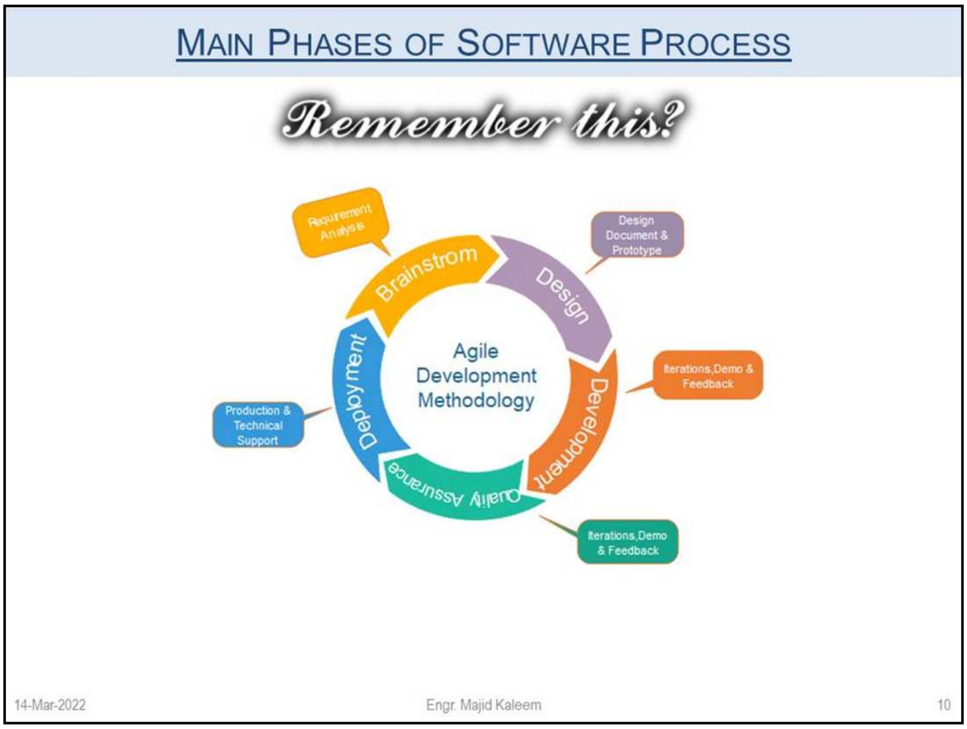
Main Phases of Software Process

Remember this?

Problem: **known**

Solution: **known**

14-Mar-2022 Engr. Majid Kaleem 9

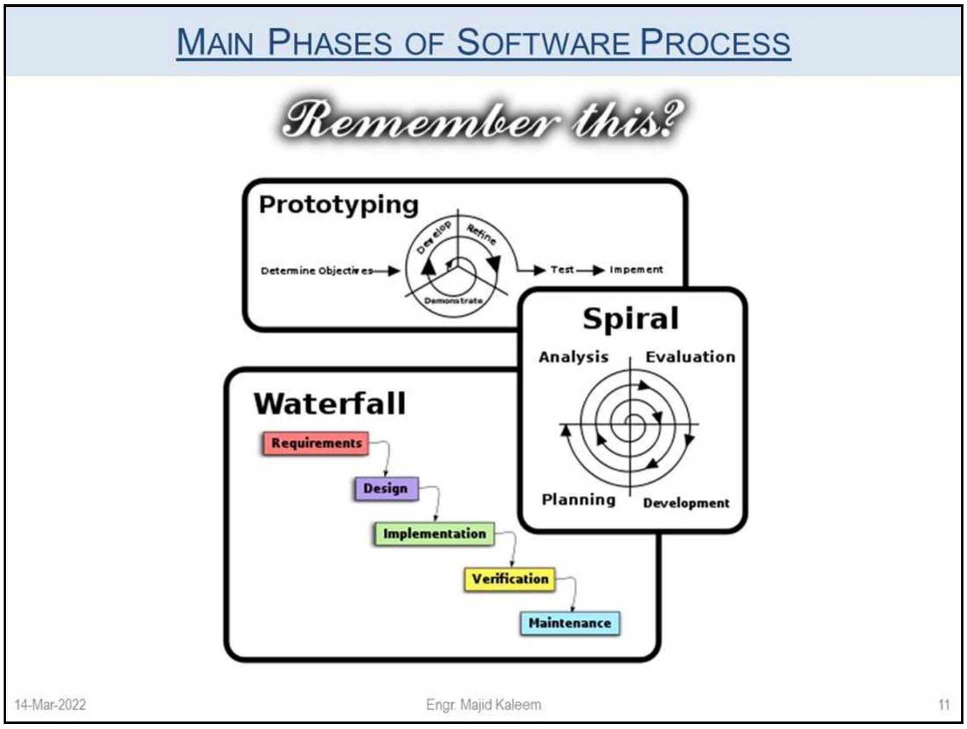


Main Phases of Software Process

Remember this?

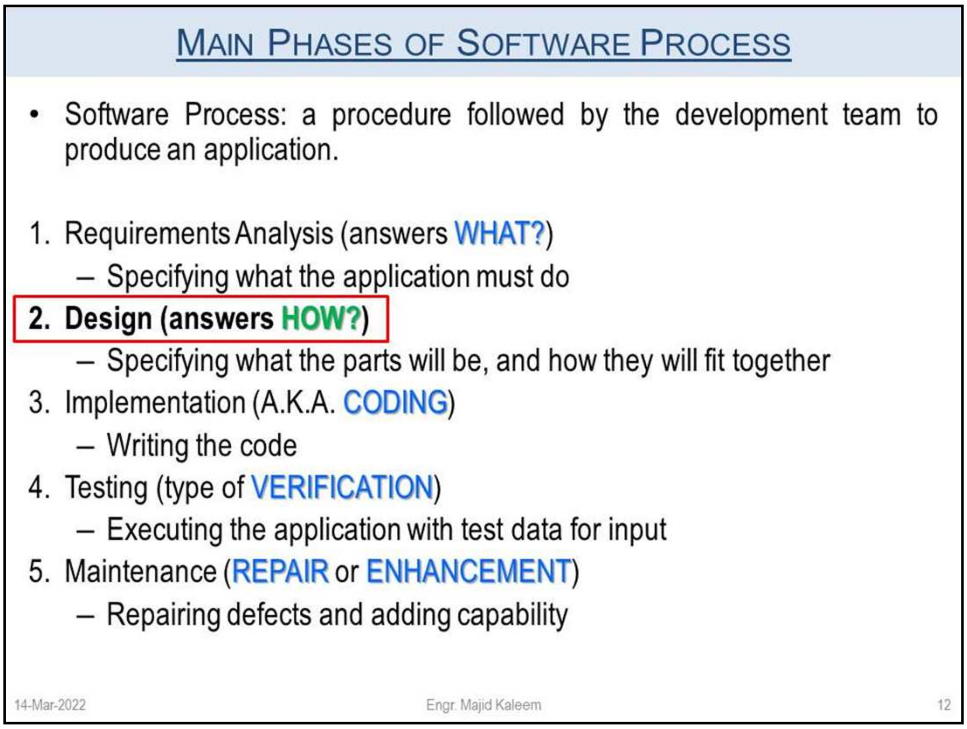
Requirement Analysis
Brainstrom
Design
Design Document & Prototype
Development
Iterations,Demo & Feedback
Quality Assurance
Iterations,Demo & Feedback
Deployment
Production & Technical Support
Agile Development Methodology

14-Mar-2022 Engr. Majid Kaleem 10



Main Phases of Software Process

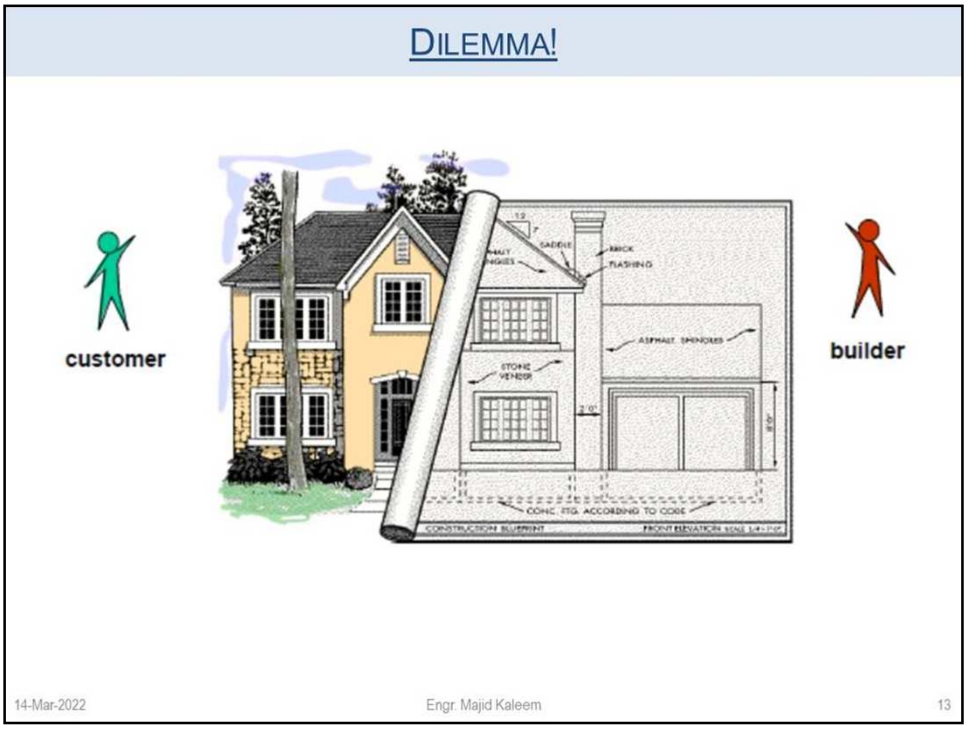
Remember this?



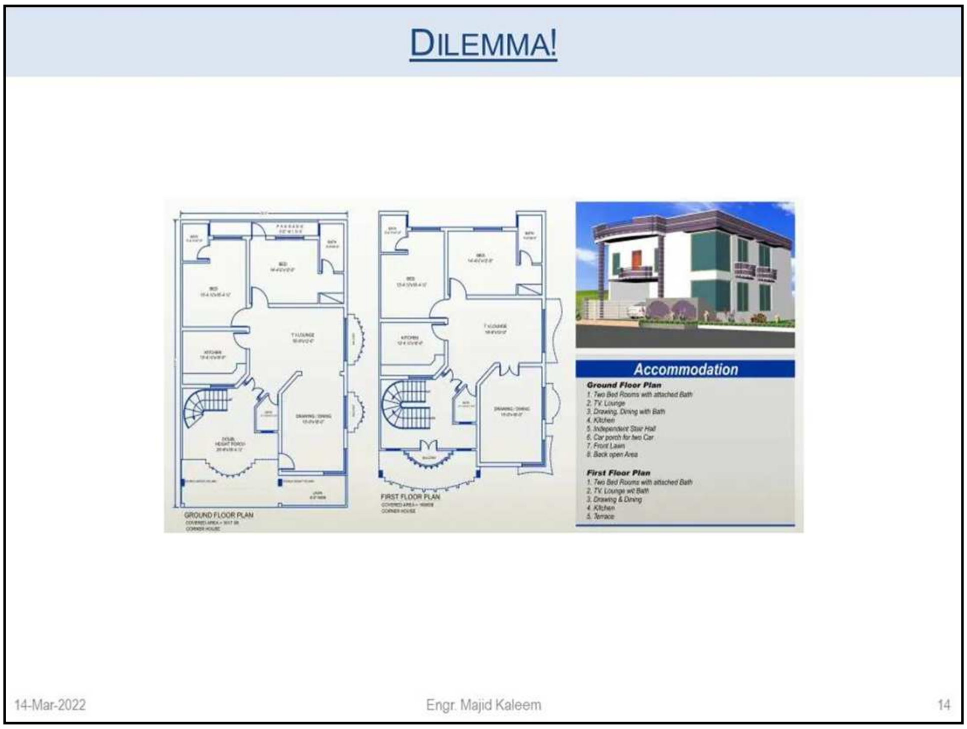
Main Phases of Software Process

- Software Process: a procedure followed by the development team to produce an application.

1. Requirements Analysis (answers WHAT?)
   - Specifying what the application must do
2. **Design (answers HOW?)**
   - Specifying what the parts will be, and how they will fit together
3. Implementation (A.K.A. CODING)
   - Writing the code
4. Testing (type of VERIFICATION)
   - Executing the application with test data for input
5. Maintenance (REPAIR or ENHANCEMENT)
   - Repairing defects and adding capability



DILEMMA!



DILEMMA!

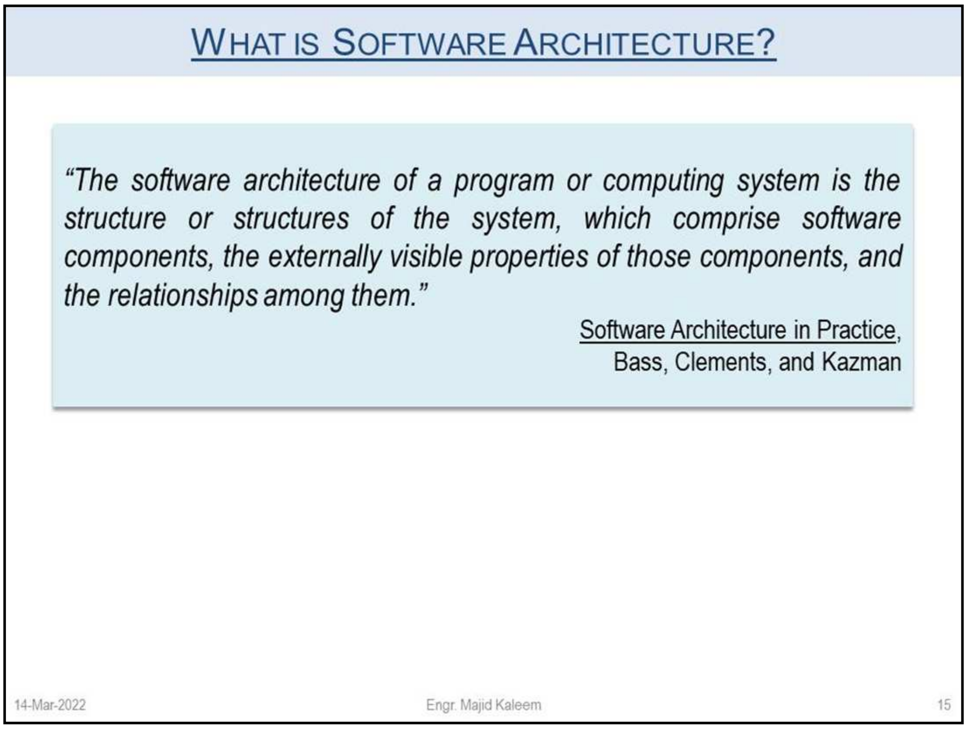
Accommodation

Ground Floor Plan

1. Two Bed Rooms with attached Bath
2. TV. Lounge
3. Drawing, Dining with Bath
4. Kitchen
5. Independent Stair Hall
6. Car porch for two Car
7. Front Lawn
8. Back open Area

First Floor Plan

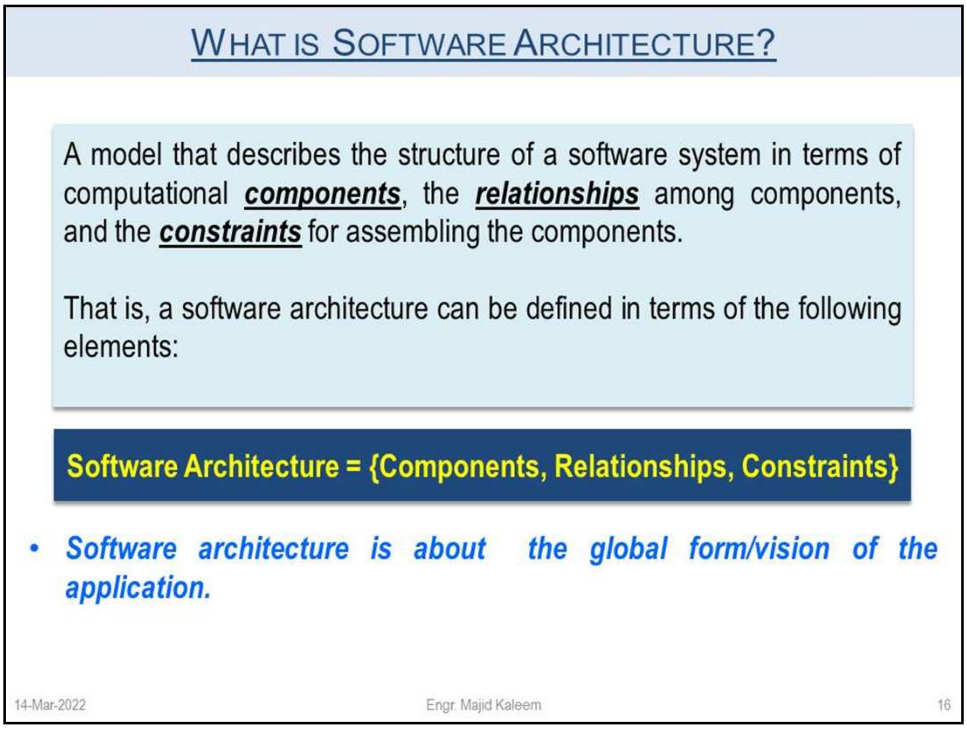
1. Two Bed Rooms with attached Bath
2. TV. Lounge w/ Bath
3. Drawing & Dining
4. Kitchen
5. Terrace



What is Software Architecture?

"The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them."

Software Architecture in Practice,
Bass, Clements, and Kazman



What is Software Architecture?

A model that describes the structure of a software system in terms of computational ***components***, the ***relationships*** among components, and the ***constraints*** for assembling the components.

That is, a software architecture can be defined in terms of the following elements:

Software Architecture = {Components, Relationships, Constraints}

- ***Software architecture is about the global form/vision of the application.***

WHAT IS SOFTWARE ARCHITECTURE?

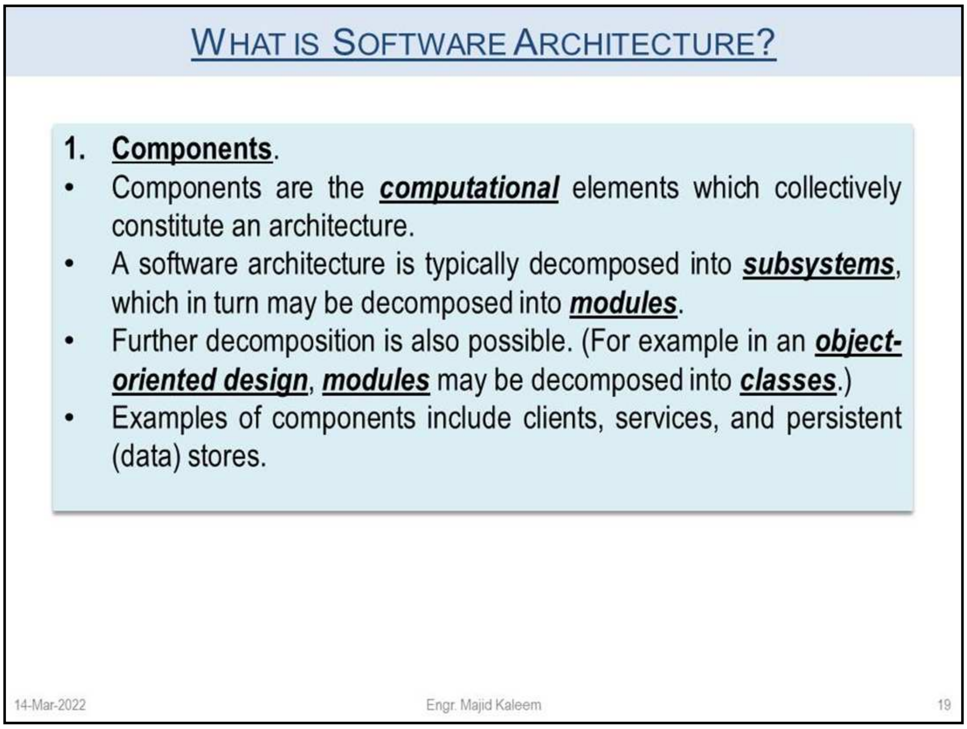
Perry and Wolf define software architecture using the following formula:

Software Architecture = {Elements, Patterns, Motivations}

- Within the context of object-oriented approach:
  - *The elements are the objects and classes,*
  - *Patterns are grouping of objects and classes,*
  - *Motivations explain why a particular grouping is better adapted than another in a given context.*

WHAT IS SOFTWARE ARCHITECTURE?

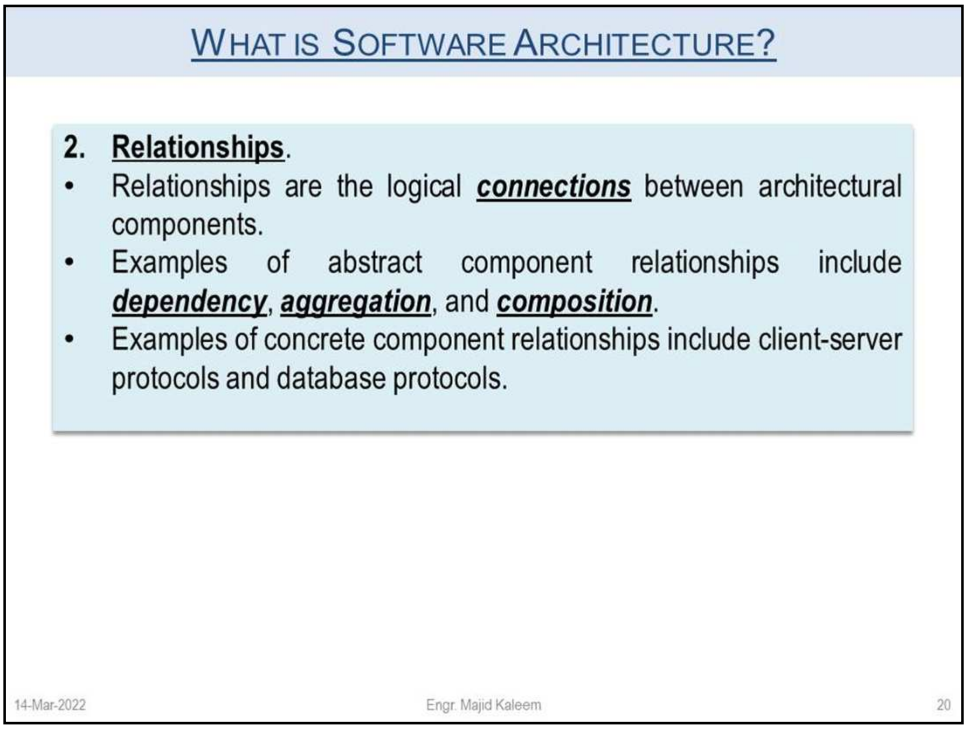
Software Architecture = {components, relationships, constraints}



WHAT IS SOFTWARE ARCHITECTURE?

1. **Components**.

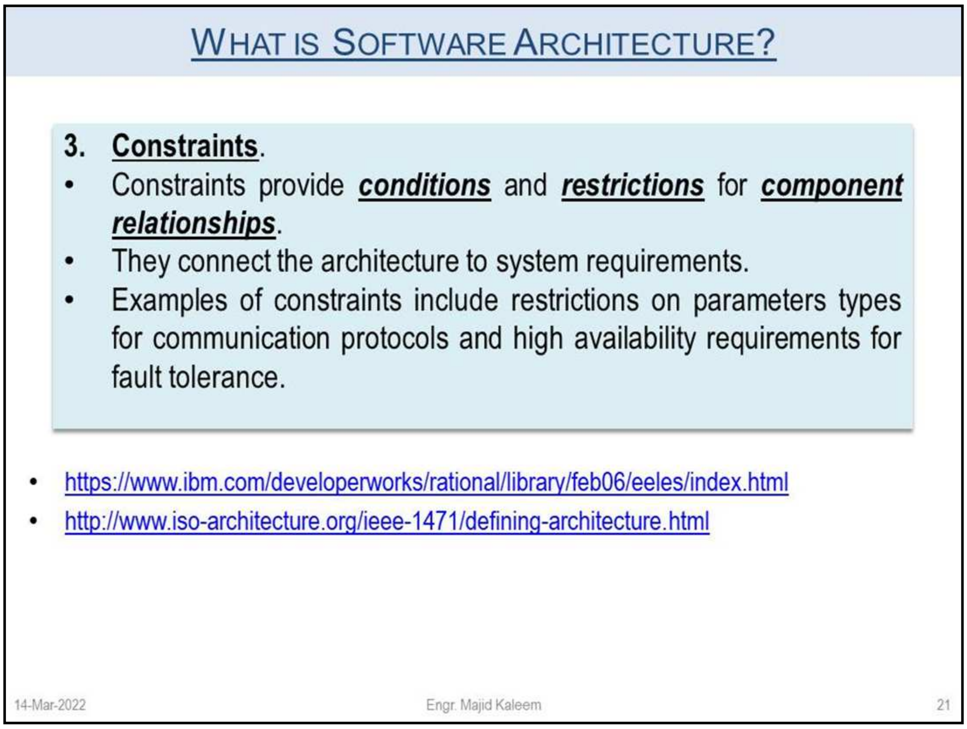
- Components are the ***computational*** elements which collectively constitute an architecture.
- A software architecture is typically decomposed into ***subsystems***, which in turn may be decomposed into ***modules***.
- Further decomposition is also possible. (For example in an ***object-oriented design***, ***modules*** may be decomposed into ***classes***.)
- Examples of components include clients, services, and persistent (data) stores.



WHAT IS SOFTWARE ARCHITECTURE?

2. **Relationships**.

- Relationships are the logical ***connections*** between architectural components.
- Examples of abstract component relationships include ***dependency***, ***aggregation***, and ***composition***.
- Examples of concrete component relationships include client-server protocols and database protocols.

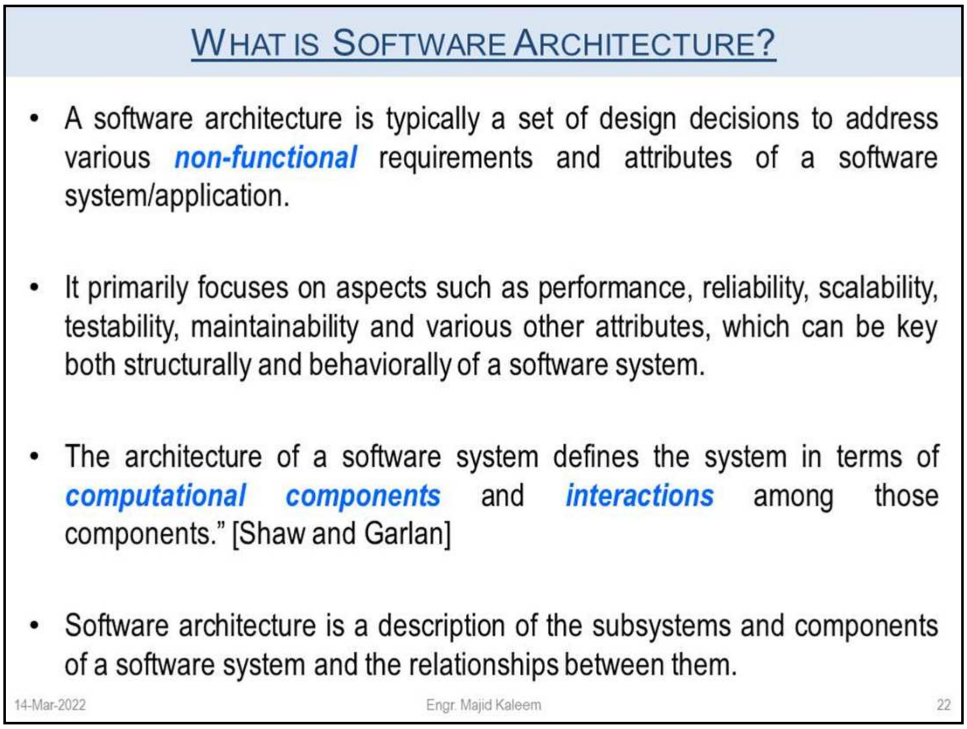


WHAT IS SOFTWARE ARCHITECTURE?

3. **Constraints**.

- Constraints provide ***conditions*** and ***restrictions*** for ***component relationships***.
- They connect the architecture to system requirements.
- Examples of constraints include restrictions on parameters types for communication protocols and high availability requirements for fault tolerance.

- https://www.ibm.com/developerworks/rational/library/feb06/eeles/index.html
- http://www.iso-architecture.org/ieee-1471/defining-architecture.html



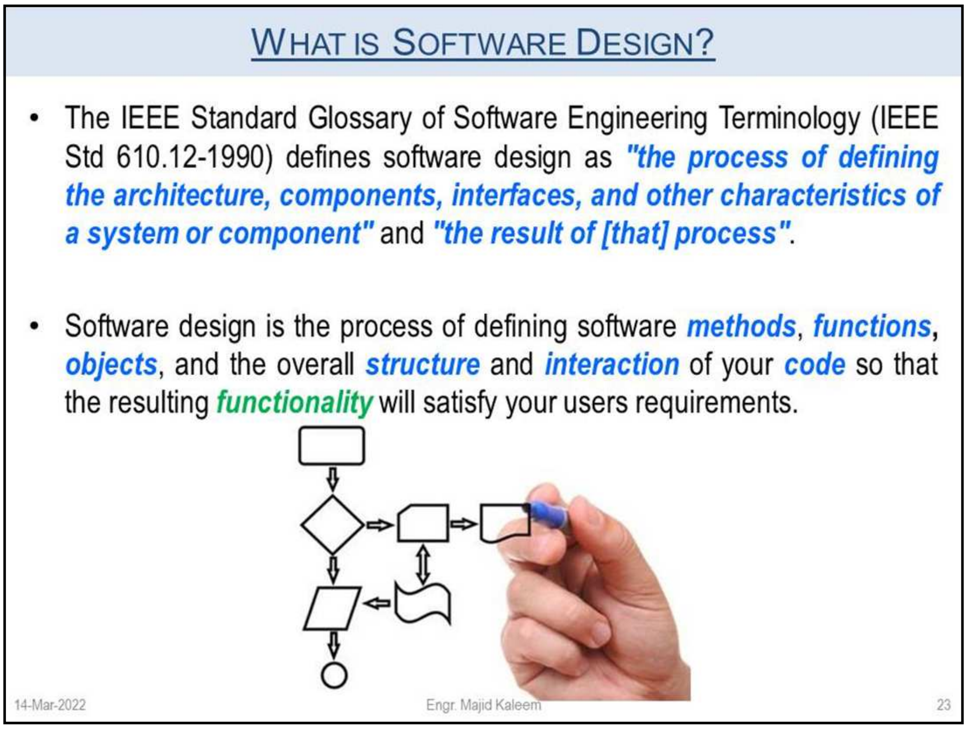
WHAT IS SOFTWARE ARCHITECTURE?

- A software architecture is typically a set of design decisions to address various ***non-functional*** requirements and attributes of a software system/application.

- It primarily focuses on aspects such as performance, reliability, scalability, testability, maintainability and various other attributes, which can be key both structurally and behaviorally of a software system.

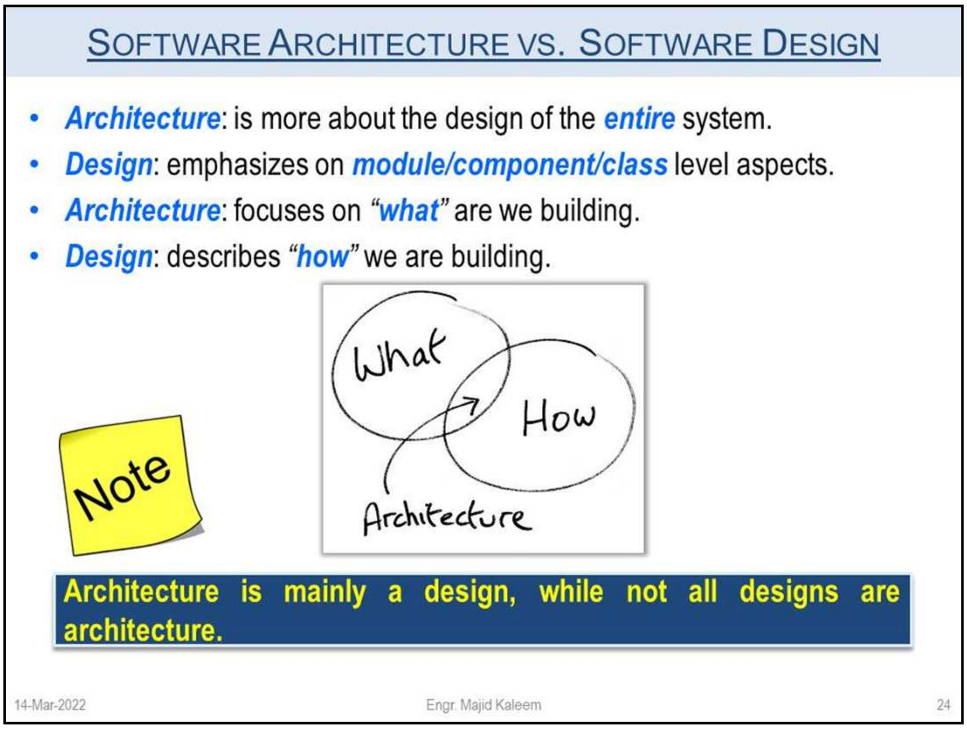
- The architecture of a software system defines the system in terms of ***computational components*** and ***interactions*** among those components." [Shaw and Garlan]

- Software architecture is a description of the subsystems and components of a software system and the relationships between them.



WHAT IS SOFTWARE DESIGN?

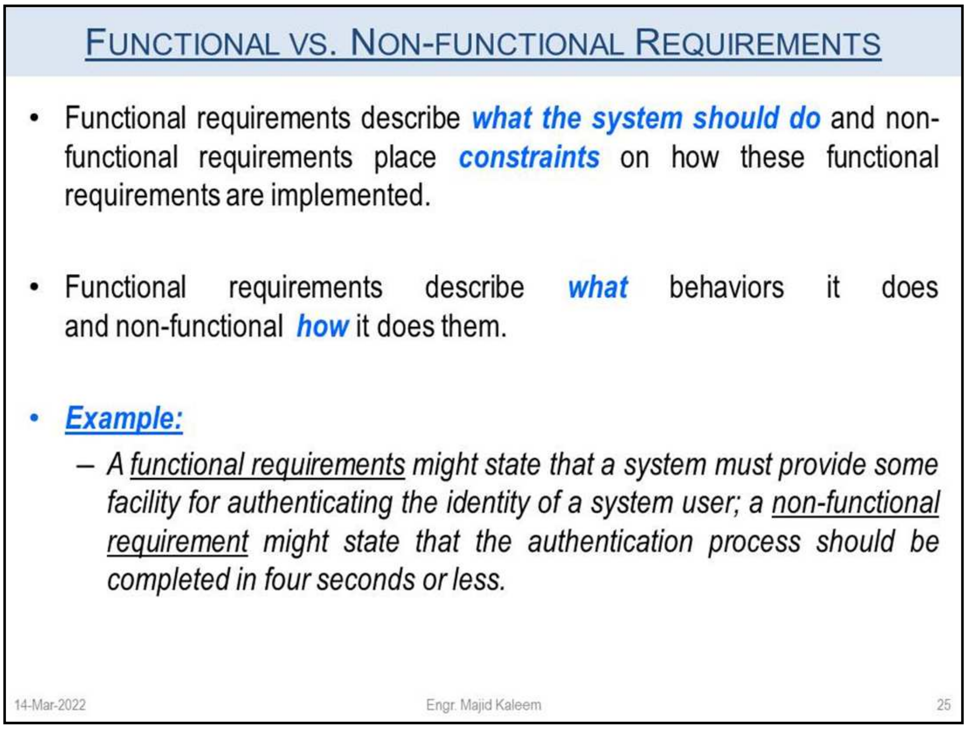
- The IEEE Standard Glossary of Software Engineering Terminology (IEEE Std 610.12-1990) defines software design as ***"the process of defining the architecture, components, interfaces, and other characteristics of a system or component"*** and ***"the result of [that] process"***.
- Software design is the process of defining software ***methods***, ***functions***, ***objects***, and the overall ***structure*** and ***interaction*** of your ***code*** so that the resulting ***functionality*** will satisfy your users requirements.



SOFTWARE ARCHITECTURE VS. SOFTWARE DESIGN

- ***Architecture***: is more about the design of the ***entire*** system.
- ***Design***: emphasizes on ***module/component/class*** level aspects.
- ***Architecture***: focuses on "***what***" are we building.
- ***Design***: describes "***how***" we are building.

Architecture is mainly a design, while not all designs are architecture.

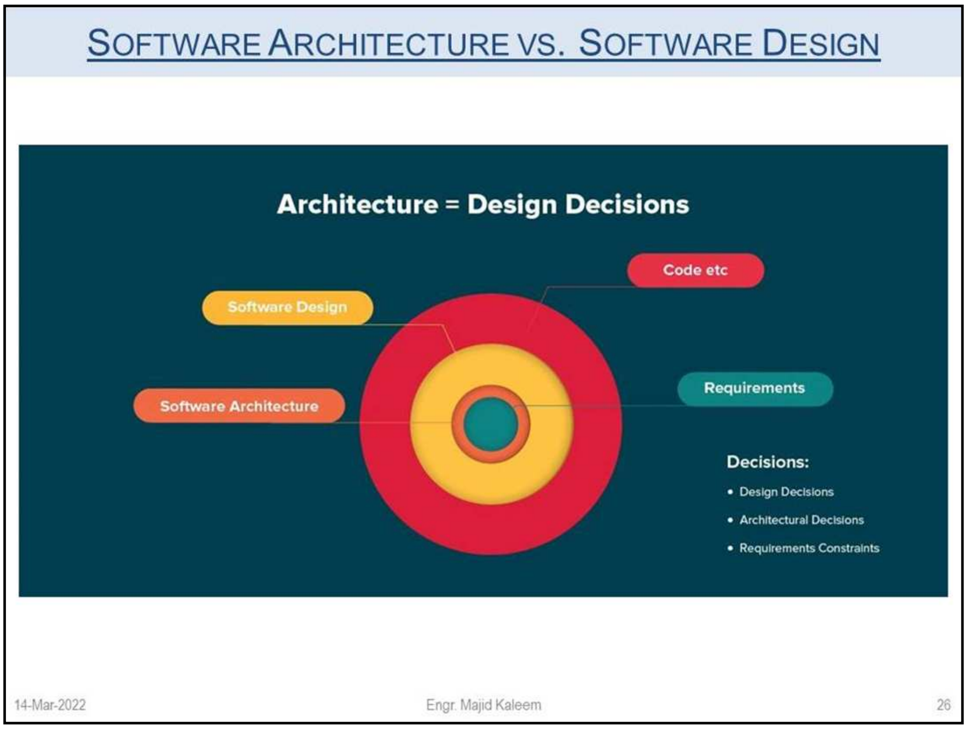


Functional vs. Non-functional Requirements

- Functional requirements describe ***what the system should do*** and non-functional requirements place ***constraints*** on how these functional requirements are implemented.

- Functional requirements describe ***what*** behaviors it does and non-functional ***how*** it does them.

- ***Example:***
  - *A functional requirements might state that a system must provide some facility for authenticating the identity of a system user; a non-functional requirement might state that the authentication process should be completed in four seconds or less.*



Software Architecture vs. Software Design

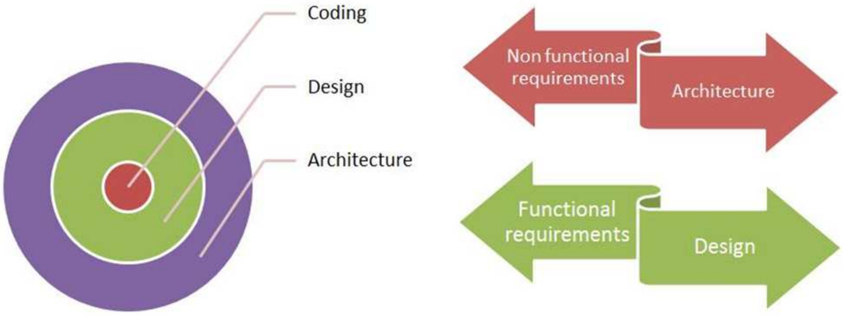
Architecture = Design Decisions

Decisions:

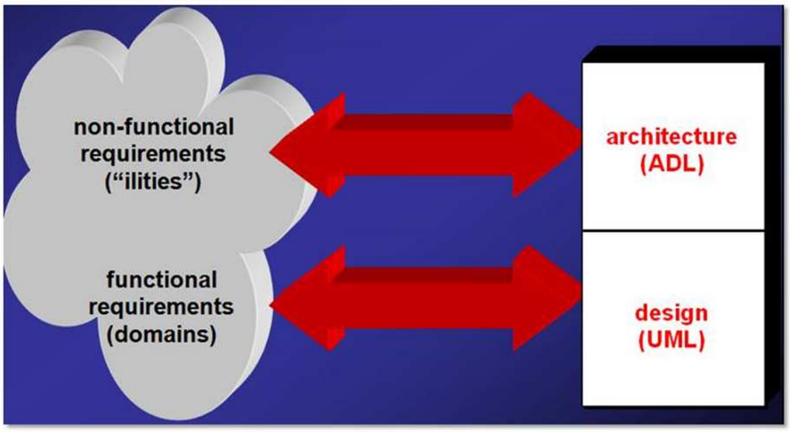
- Design Decisions
- Architectural Decisions
- Requirements Constraints

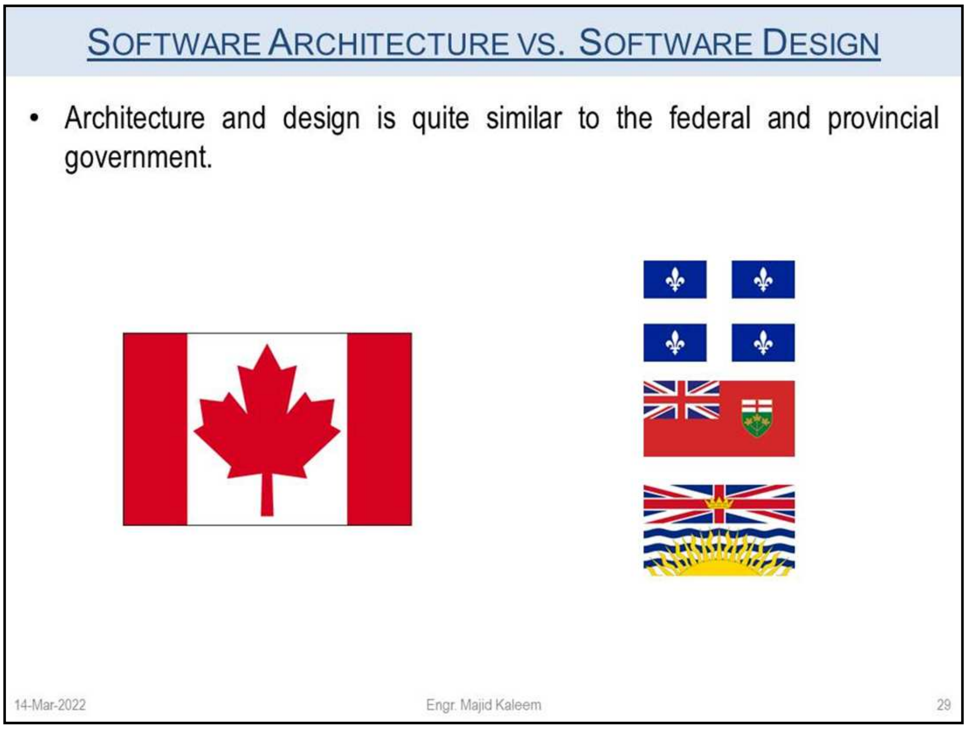
SOFTWARE ARCHITECTURE VS. SOFTWARE DESIGN

- ***Architecture:*** where non-functional decisions are cast, and functional requirements are partitioned
- ***Design:*** where functional requirements are accomplished.



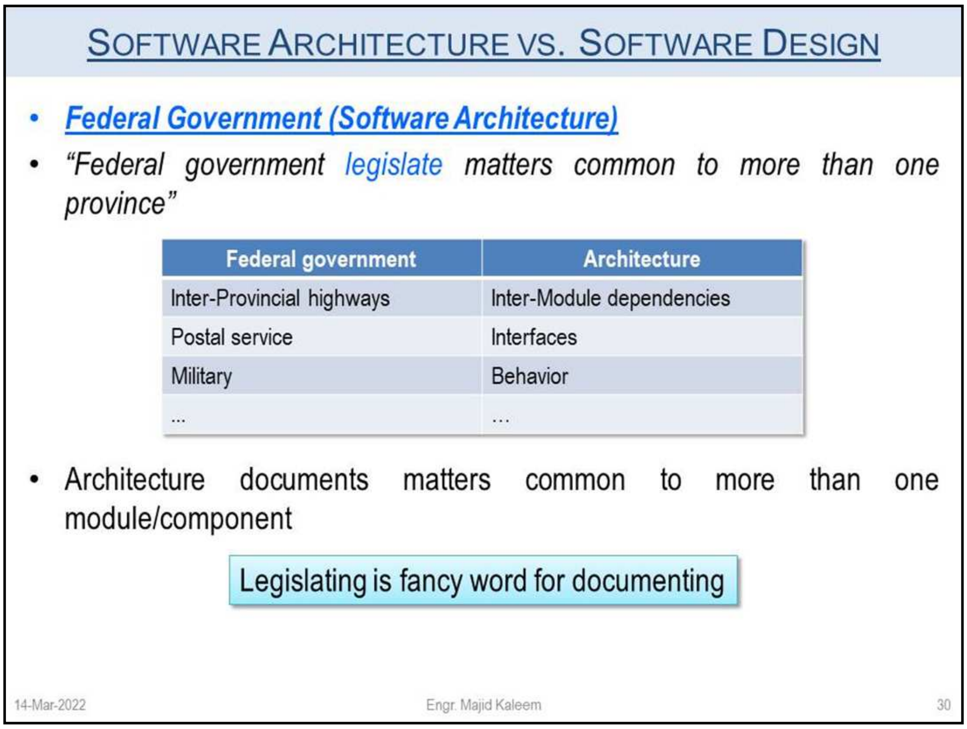
SOFTWARE ARCHITECTURE VS. SOFTWARE DESIGN


non-functional requirements ("ilities")
functional requirements (domains)
architecture (ADL)
design (UML)



SOFTWARE ARCHITECTURE VS. SOFTWARE DESIGN

- Architecture and design is quite similar to the federal and provincial government.



SOFTWARE ARCHITECTURE VS. SOFTWARE DESIGN

- ***Federal Government (Software Architecture)***
- *"Federal government legislate matters common to more than one province"*

Federal government	Architecture
Inter-Provincial highways	Inter-Module dependencies
Postal service	Interfaces
Military	Behavior
...	…

- Architecture documents matters common to more than one module/component

Legislating is fancy word for documenting

SOFTWARE ARCHITECTURE VS. SOFTWARE DESIGN

- ***Provincial Government (Software Design)***
- *"Provinces may legislate on matters of a merely local or private nature"*

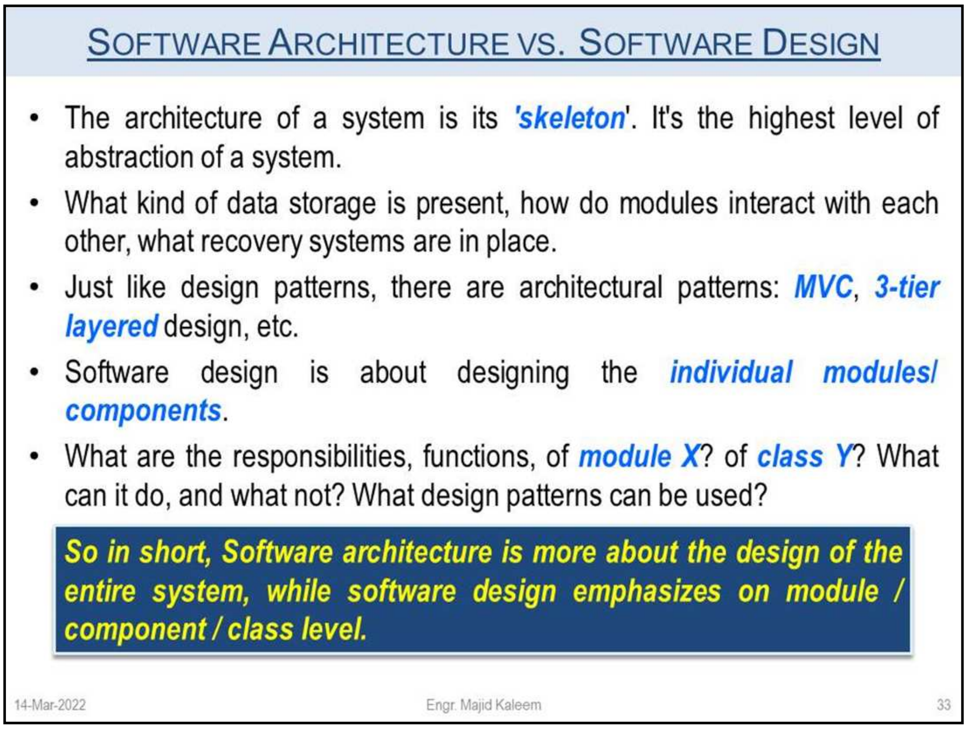
Provincial government	Software design
Education	Classes and objects
Provincial officers	Software design patterns
Municipal government	Dependencies
...	...

- Software design may document on matters of a merely local or private nature

"may" because you don't need to document everything

SOFTWARE ARCHITECTURE VS. SOFTWARE DESIGN

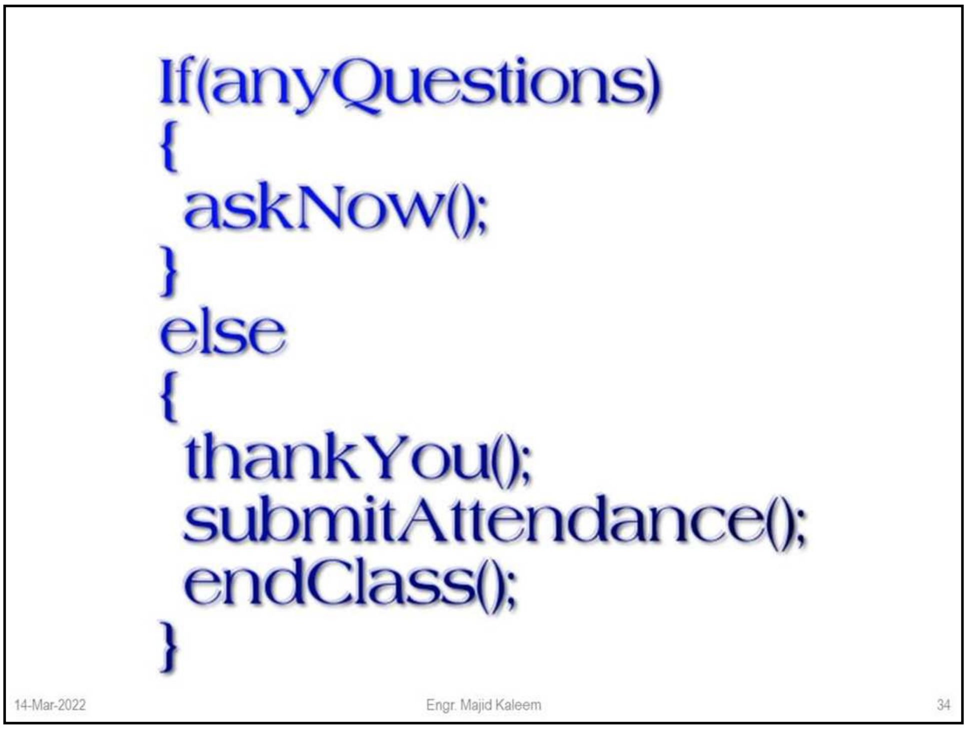
EXAMPLE



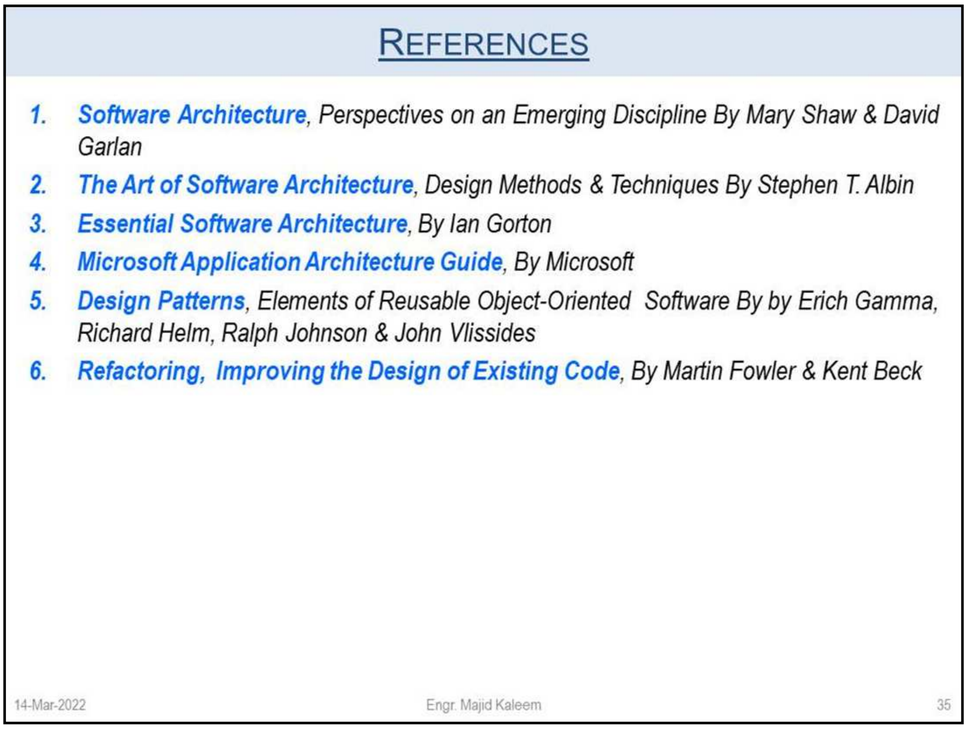
Software Architecture vs. Software Design

- The architecture of a system is its ***'skeleton'***. It's the highest level of abstraction of a system.
- What kind of data storage is present, how do modules interact with each other, what recovery systems are in place.
- Just like design patterns, there are architectural patterns: ***MVC***, ***3-tier layered*** design, etc.
- Software design is about designing the ***individual modules/ components***.
- What are the responsibilities, functions, of ***module X***? of ***class Y***? What can it do, and what not? What design patterns can be used?

So in short, Software architecture is more about the design of the entire system, while software design emphasizes on module / component / class level.



```
If(anyQuestions)
{
 askNow();
}
else
{
 thankYou();
 submitAttendance();
 endClass();
}
```



REFERENCES

1. ***Software Architecture***, *Perspectives on an Emerging Discipline By Mary Shaw & David Garlan*
2. ***The Art of Software Architecture***, *Design Methods & Techniques By Stephen T. Albin*
3. ***Essential Software Architecture***, *By Ian Gorton*
4. ***Microsoft Application Architecture Guide***, *By Microsoft*
5. ***Design Patterns***, *Elements of Reusable Object-Oriented Software By by Erich Gamma, Richard Helm, Ralph Johnson & John Vlissides*
6. ***Refactoring, Improving the Design of Existing Code***, *By Martin Fowler & Kent Beck*

14-Mar-2022 Engr. Majid Kaleem 35