Unit 3 – Software Design

1. Design Process

- Iterative: SRS → high-level design → detailed design → code

2. Core Principles

- Traceability to requirements
- Reusability & avoid reinventing
- Minimize intellectual distance (mirror real world)
- Modularity, change-tolerance, graceful degradation
- Continuous quality checks

3. Abstraction & Refinement

- **Abstraction**: hide low-level details
- **Refinement**: stepwise decomposition to code

4. UML Views & Diagrams

View	Focus	Key Diagrams
User	Actor/scenarios	Use-case
Structural	Classes/objects	Class, Object
Behavioral	Interactions	Sequence, Collaboration, State, Activity
Implementation	Code modules	Component
Environment	Deployment	Deployment

5. Architectural Styles

- **Data-Centered**: repo / blackboard
- **Data-Flow**: pipes & filters, batch seq
- **Call-&-Return**: layered modules
- **Object-Oriented**: objects + messages
- **Layered**: UI → App → Util → Core

6. 4+1 “Views” Model

- Logical, Development, Process, Physical + Scenarios

7. UI Design Fundamentals

- Goals: attractive, simple, responsive, consistent
- Principles: Structure, Simplicity, Visibility, Feedback, Tolerance, Reuse
- Flow: tasks → prototype → implement → evaluate

8. Design Paradigms

- **Function-Oriented**: DFDs → charts → PDL
- **Component-Based**: reusable components w/ clear interfaces

9. Design Metrics

- Architectural: module count, connector complexity
- OO: size, coupling, cohesion, complexity, volatility
- UI: layout appropriateness, screen cohesion