<table>
<thead>
<tr>
<th>Unit</th>
<th>Task</th>
<th>No of Hours Required</th>
<th>Actual hours Taken, Reason for the deviation</th>
<th>Teaching Method</th>
<th>Book</th>
<th>Dependencies (page no.)</th>
<th>Remarks</th>
<th>No of Students Present</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIT I (9 hours)</td>
<td>Introduction</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td></td>
<td>8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>What is software architecture</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td>11,14</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Standard Definitions- Architectural structures</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td>24,27</td>
<td>ASSIGNMENT-I</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Influence of software architecture on organization both business and technical</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td>31,33</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Architecture Business Cycle</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td>48,52</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Introduction - Functional requirements</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td>7,19</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Technical constraints</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td>62</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Quality Attributes</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td>74</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Lesson Plan & Schedule (Sub.Code: IT6602 – Sub Name: Software Architecture)

<table>
<thead>
<tr>
<th>Unit</th>
<th>Task</th>
<th>No of Hours Required</th>
<th>Actual hours Taken, Reason for the deviation</th>
<th>Teaching Method</th>
<th>Book</th>
<th>Dependencies (page no.)</th>
<th>Remarks</th>
<th>No of Students Present</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIT II</td>
<td>Quality Attribute Workshop</td>
<td>1</td>
<td>BB T1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(9 hours)</td>
<td>Quality attribute Scenario-</td>
<td>1</td>
<td>BB T1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Availability Scenario</td>
<td>1</td>
<td>BB T1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Modifiability</td>
<td>1</td>
<td>BB T1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Security</td>
<td>1</td>
<td>BB T1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Performance-Testability-Useability</td>
<td>1</td>
<td>BB T1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ASSIGNMENT-II</td>
</tr>
<tr>
<td></td>
<td>Documenting Quality Attributes</td>
<td>1</td>
<td>BB T1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Six part scenarios</td>
<td>1</td>
<td>BB T1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Case studies</td>
<td>1</td>
<td>BB T1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **UNIT II (9 hours)**
  - Quality Attribute Workshop: 1 hour
  - Quality attribute Scenario: 1 hour
  - Availability Scenario: 1 hour
  - Modifiability: 1 hour
  - Security: 1 hour
  - Performance-Testability-Useability: 1 hour
  - Documenting Quality Attributes: 1 hour
  - Six part scenarios: 1 hour
  - Case studies: 1 hour

- **Remarks**
  - 105,110
  - 115,122
  - 163,166,169
  - 176
  - 203
  - 206,212
  - 213,223
  - 261
### Lesson Plan & Schedule (Sub.Code: IT6602 – Sub Name: Software Architecture)

<table>
<thead>
<tr>
<th>Unit</th>
<th>Task</th>
<th>No of Hours Required</th>
<th>Actual hours Taken, Reason for the deviation</th>
<th>Teaching Method</th>
<th>Book</th>
<th>Dependencies (page no.)</th>
<th>Remarks</th>
<th>No of Students Present</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Unit III</strong></td>
<td><strong>9 Hours</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Introduction</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td>351</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Standard Definitions for views</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td>360</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Structures and views</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td>366</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Designing Integration Tests – Integration Test Planning Scenario testing</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td>397</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Representing views</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td>401</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>available notations</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td>409</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Standard views</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td>421</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4+1 view of RUP- Siemens 4 views</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td>425,436</td>
<td>ASSIGN MENT-III</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SEI’s perspectives and views – Case studies</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td>445</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit</td>
<td>Task</td>
<td>No of Hours Required</td>
<td>Actual hours Taken, Reason for the deviation</td>
<td>Teaching Method</td>
<td>Book</td>
<td>Dependencies (page no.)</td>
<td>Remarks</td>
<td>No of Students Present</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>----------------------</td>
<td>---------------------------------------------</td>
<td>----------------</td>
<td>------</td>
<td>------------------------</td>
<td>---------</td>
<td>------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>BB/PP/Video</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Unit IV (9 Hours)**

<table>
<thead>
<tr>
<th>Task</th>
<th>No of Hours Required</th>
<th>Actual hours Taken, Reason for the deviation</th>
<th>Teaching Method</th>
<th>Book</th>
<th>Dependencies (page no.)</th>
<th>Remarks</th>
<th>No of Students Present</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td></td>
<td>467</td>
<td></td>
</tr>
<tr>
<td>Basic Properties</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td></td>
<td>472,478</td>
<td></td>
</tr>
<tr>
<td>Data flow styles</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td></td>
<td>503</td>
<td></td>
</tr>
<tr>
<td>Pipe and Filter</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td></td>
<td>515,528</td>
<td></td>
</tr>
<tr>
<td>Call-return styles</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td></td>
<td>543</td>
<td>ASSIGNMENT-IV</td>
</tr>
<tr>
<td>Layered system</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td></td>
<td>515</td>
<td></td>
</tr>
<tr>
<td>Shared Information styles</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td></td>
<td>553</td>
<td></td>
</tr>
<tr>
<td>Event styles</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td></td>
<td>562</td>
<td></td>
</tr>
</tbody>
</table>
Lesson Plan & Schedule (Sub.Code: IT6602 - Sub Name: Software Architecture)

<table>
<thead>
<tr>
<th>Unit</th>
<th>Task</th>
<th>No of Hours Required</th>
<th>Actual hours Taken, Reason for the deviation</th>
<th>Teaching Method BB/PP/Video</th>
<th>Book</th>
<th>Dependencies (page no.)</th>
<th>Remarks</th>
<th>No of Students Present</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>Good practices</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td></td>
<td>632</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>Documenting the Views using UML</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td></td>
<td>635</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>Merits and Demerits of using visual languages</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td></td>
<td>672</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>Need for formal languages</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td></td>
<td>677</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>Architectural Description Languages</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td></td>
<td>684</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>ACME</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td></td>
<td>685</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>Case studies</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td></td>
<td>699</td>
<td>ASSIGNMENT-V</td>
</tr>
<tr>
<td>V</td>
<td>Special topics: SOA and Web services</td>
<td>1</td>
<td></td>
<td>BB</td>
<td>T1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Name of the Department: INFORMATION TECHNOLOGY
Name of the Faculty: Mr. R. CRISTIN

Year /Semester: III/VI
Lesson Plan & Schedule (Sub.Code: IT6602- Sub Name: Software Architecture)

<table>
<thead>
<tr>
<th>Sl.no</th>
<th>Author</th>
<th>Title</th>
<th>Publication</th>
</tr>
</thead>
</table>

(Note: BB= Black Board, PP=Power Point Presentation, VIDEO = Video Lecture)
Name of the Department: INFORMATION TECHNOLOGY
Year /Semester: III/VI

Lesson Plan & Schedule (Sub.Code: IT6602 – Sub Name: Software Architecture)

Signature of the Staff

Signature of the HOD

Signature of Principal
IT6602 SOFTWARE ARCHITECTURES
UNIT I
INTRODUCTION AND ARCHITECTURAL DRIVERS


COURSE OBJECTIVE: Understand software architectural requirements and drivers

PART- A

1. **What is software architecture?**

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

2. **What is architectural pattern?**

   An architectural pattern is a description of element and relation types together with a set of constraints on how they may be used.

3. **Compare System Architecture and Software Architecture.**

   - **Software architecture** focuses on the implementation that will solve a specific problem.
   - **Systems architecture** focuses on the underlying servers (physical or virtual) and server software (web servers, database et cetera) that the software will utilize. Systems architecture is largely focused on scaling with high availability, fault tolerance and redundancy to avoid data loss.

4. **What is reference model?**

   A reference model is a division of functionality together with data flow between the pieces. A reference model is a standard decomposition of a known problem into parts that cooperatively solve the problem.
5. What is reference architecture?

A reference architecture is a reference model mapped onto software elements (that cooperatively implement the functionality defined in the reference model) and the data flows between them. Whereas a reference model divides the functionality, a reference architecture is the mapping of that functionality onto a system decomposition.

6. Why is software architecture important?

- Architecture is the vehicle for stakeholder communication
- Architecture manifests the earliest set of design decisions
- Architecture as a transferable, re-usable model

These are the reasons for the importance of software architecture from the technical perspective.

7. What is Architecture Business Cycle?

Software architecture is a result of technical, business, and social influences. Its existence in turn affects the technical, business, and social environments that subsequently influence future architectures. We call this cycle of influences, from the environment to the architecture and back to the environment, the Architecture Business Cycle (ABC).

8. List out the requirements for Architecture Business Cycle.

- Case studies
- Methods
- Techniques

9. What are the three classes of influence that come from the developing organization?

- Immediate business
- Long-term business
- Organizational structure.

10. What activities are involved in creating software architecture?

- Creating the business case for the system
- Understanding the requirements
Creating or selecting the architecture
Documenting and communicating the architecture
Analyzing or evaluating the architecture
Implementing the system based on the architecture
Ensuring that the implementation conforms to the architecture

11. What is a variety of techniques for understanding requirements from the stakeholders?

There are a variety of techniques for eliciting requirements from the stakeholders:
- Object-oriented analysis uses scenarios, or "use cases" to represent requirements.
- Collection of quality attribute scenarios that support the capture of quality requirements for a system.
- Another technique that helps us understand requirements is the creation of prototypes.

12. What is architectural structures and its types?

A structure is the set of elements itself, as they exist in software or hardware. The types of architectural structures are
- Module structures
- Component-and-connector structures
- Allocation structures

13. What are Quality attribute scenarios?

Quality attribute scenarios are the means by which quality moves from the eye of the beholder to a more objective basis.

14. Define functional requirements and list its 2 characteristics. (MAY/JUNE 2016)

- A functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behavior, and outputs.
- Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish.
- A functional requirement describes what a software system should do.

Some of the characteristics of functional requirements are:
Business Rules
- Transaction corrections, adjustments and cancellations
- Administrative functions
- Authentication
- Authorization levels
- Audit Tracking
- External Interfaces

15. Write various technical constraints and business constraints.
(MAY/JUNE 2016) Technical Constraints in Software Architecture

Technical constraints are fixed technical design decisions that absolutely cannot be changed.
- Programming language
- Operating system or platforms supported
- Use of a specific library or framework

Business Constraints
Business constraints are unchangeable business decisions that in some way restrict the software architecture design.
- Schedule
- Budget
- Team composition and make-up
- Software licensing restrictions or requirements

16. What is quality attributes?
A quality attributes is a property of a process or product that can have some quantitative value and can be measured or observed.

17. What is functionality?
Functionality is the ability of the system to do the work for which it was intended.

18. Which is quality attributes challenging?
Quality attributes often impact the system as a whole.
It interrelates with functionality.
It often in conflict with each other.
19. **List the various classes of quality attributes?**

Qualities of the system  
Business qualities  
Qualities of the architecture itself.

20. **Define the term perspective?**

Perspective contain proven architectural knowledge and help structure the architecture definition process by separating concerns but focusing on cross-structured quality properties rather than architectural structures.

**UNIT II**  
**QUALITY ATTRIBUTE WORKSHOP**

**SYLLABUS:** Quality Attribute Workshop – Documenting Quality Attributes – Six part scenarios – Case studies.

**COURSE OBJECTIVE:** To understand the quality attributes for software architecture.

**PART- A**

1. **What is software quality?**

Software **quality** is the degree to which software possesses a desired combination of attributes.

2. **What are the properties a critical system must satisfy?**

Critical systems must satisfy security, safety, dependability, performance, and other, similar requirements as well.

3. **What is functionality?**

It is the ability of the system to do the work for which it was intended.
4. What are the six part scenarios of a quality attribute workshop?

QAW elicits and records six-part scenarios, where the parts include the stimulus of the scenario, the source of the stimulus, the response, the response measure, the artifact stimulated, and the environment.

5. What are the inputs to QAW? Inputs include the

- System’s business/mission drivers
- System’s architectural plan

6. What are the steps of QAW?

QAW method includes the following steps:
- Business/Mission Presentation: Architectural Plan Presentation:
- Identification of Architectural Drivers
- Scenario Brainstorming
- Scenario Consolidation
- Scenario Prioritization
- Scenario Refinement

7. What are the outputs of QAW? (OR) List QAW Results? (MAY/JUNE 2016)

Outputs include a list of
- Raw scenarios
- Consolidated scenarios
- Prioritized scenarios
- Refined scenarios

8. What is quality attribute scenario?

A quality attribute scenario is a quality-attribute-specific requirement. It consists of six parts.
- Source of stimulus.
- Stimulus.
- Environment.
- Artifact
- Response.
- Response measure
9. What is security scenario?

Security is a measure of the system's ability to resist unauthorized usage while still providing its services to legitimate users.

10. What is software testability?

Software testability refers to the ease with which software can be made to demonstrate its faults through testing. In particular, testability refers to the probability, assuming that the software has at least one fault that it will fail on its next test execution.

11. What is usability scenario?

Usability is concerned with how easy it is for the user to accomplish a desired task and the kind of user support the system provides.

12. What is modifiability?

Modifiability is about the cost of change. It brings up two concerns.

☐ What can change (the artifact)?
☐ When is the change made and who makes it (the environment)?

13. List out various business and architectural qualities. (MAY/JUNE 2016)

**BUSINESS QUALITIES**
- Time to market.
- Cost and benefit.
- Projected lifetime of the system.
- Targeted market.
- Rollout schedule.
- Integration with legacy systems.

**ARCHITECTURE QUALITIES**
- Conceptual integrity
- Correctness and completeness
- Buildability
14. What is meant by functionality and architecture? (MAY/JUNE 2016)

**Functionality:**

It is the ability of the system to do the work for which it was intended. **Software architecture** constrains its allocation to structure when other quality attributes are important.

15. What are the benefits of QAW?

- Clarified quality attribute requirements
- Improved architecture documentation
- Identified risks early in the life-cycle

16. Describe term Scenario?

A scenario is a short statement describing an interaction of one of the stakeholders with the system. An architectural scenario is a crisp, concise description of a situation that the system is likely to face in its production environment, along with the definition of the system.

17. What are the features of concern scenario?

A collection of concrete scenarios can be used as the quality attribute aspect to the system in requirement of the system.

- Each scenario is concrete enough to be meaningful to the architect.
- The details of the responses are meaningful enough so that it is possible to test whether the system has achieved the response.

18. Define security scenario?

Security is measure of the system’s ability to resist unauthorized usage while providing its services to legitimate users. An attempt to breach security is an attack; it could be to gain access to data or services or to deny services to others.

19. What is use of scenario brainstorming?

After the architectural drivers have been identified, the facilitators initiate the brainstorming process in which stakeholders generate scenarios. The facilitators review the parts of a good scenario and ensure a scenario is well formed.
during the workshop. Each stakeholder expresses a scenario representing his or her concerns with round-robin fashion.

20. What are the parameters of security scenario?

- Confidentiality
- Integrity
- Non-repudiation
- Authentication
- Auditing


The tactics used by the architect to create a design using design patterns, architectural strategies.

A tactic is a design decision that influences the control of a quality attribute responses.

UNIT III

ARCHITECTURAL VIEWS


COURSE OBJECTIVE: Be exposed to various architectural views.

PART - A

1. What is view?

A view simply represents a set of system elements and relationships among them, so whatever elements and relationships you deem useful to a segment of the stakeholder community constitute a valid view.

2. What are the categories of architectural structures?

Architectural structures are divided into three groups: module, component-and-connector (C&C), and allocation.

3. What is the procedure for choosing the views for your project?
Produce a candidate view list
Combine views
Prioritize

4. What are the parts of a documented view?
- Primary presentation
- Element catalog
- Context diagram
- Variability guide
- Architecture background
- Glossary of terms

5. What are the main types of views?
- Logical
- Development
- Physical

6. List out the view in Siemens approach?
- Conceptual architecture view
- Module architecture view
- Execution architecture view
- Code architecture view

7. What is 4+1 view of RUP?
- Logical
- Process.
- Development
- Physical
- Scenarios (usecases)
8. Distinguish structures and views.

☐ A view is a representation of a coherent set of architectural elements, as written by and read by system stakeholders. It consists of a representation of a set of elements and the relations among them.

☐ A structure is the set of elements itself, as they exist in software or hardware. For example, a module structure is the set of the system's modules and their organization. A module view is the representation of that structure, as documented by and used by some system stakeholders.


A style is a specialization of element types (e.g., “client,” “layer”) and relationship types (e.g., “is part of,” “request-reply connection,” “is allowed to use”), along with any restrictions (e.g., “clients interact with servers but not each other” or “all the software comprises layers arranged in a stack such that each layer can only use software in the next lower layer”).

10. List out the sections in cross view or beyond views documentation.

Cross-view or “beyond views” documentation consists of the following sections:

☐ Documentation roadmap
☐ View template.
☐ System overview
☐ Mapping between views.
☐ Directory.
☐ Project glossary and acronym list.
☐ Cross-view rationale.

11. Define a viewpoint. (MAY/JUNE 2016)

A viewpoint is a collection of patterns, templates, and conventions for constructing one type of view. It defines the stakeholders whose concerns are reflected in the viewpoint and the guidelines, principles, and template models for constructing its views.

12. How to choose views. (MAY/JUNE 2016)

This is a three-step procedure for choosing the views for your project:

☐ Produce a candidate view list:
☐ Combine views:
Prioritize:

13. **What is view point?**
   
   A view point is a collection of patterns, templates, and conventions for constructing one type of view. It defines the stakeholders whose concerns are reflected in the view points and guidelines, principles and templates models for constructing its views.

14. **Define the term Concern**
   
   A concern about architecture is a requirement, an objective, an intention, or an aspiration a stakeholder has for that architecture.

15. **What is an architectural element?**
   
   An architectural element is a fundamental piece from which a system can be considered to be constructed.

16. **What is an Interface?**
   
   An interface is a boundary across the two independent entities meet and interacts or communicates with each other.

17. **How architects examine the system?**
   
   An architect examine the system in three ways
   
   15. Module View type
   16. Components and connectors view type
   17. Allocation view type

18. **How RUP helps in developing system?**
   
   RUP is a software engineering process, which provides a disciplined approach in order to assign the tasks and responsibilities in a development organization, placing particular emphasis on architecture through its 4+1 views.

19. **Explain use of view and view point?**
   
   View and viewpoints display related concepts simultaneously.
Both are support displaying the same data at various levels of abstractions
They can be adapted to the needs of specific stakeholders.

20. What are the limitations of viewpoints?

- Inconsistency: Using a number of views to describe a system inevitably brings consistency problems
- Selection of the wrong set of view: It is not always obvious which set of views is suitable for describing a particular system.

UNIT IV
ARCHITECTURAL STYLES

SYLLABUS: Introduction – Data flow styles – Call-return styles – Shared Information styles - Event styles – Case studies for each style.

COURSE OBJECTIVE: Be exposed to various architectural styles.

PART - A

1. What is architectural style?

An architectural style, sometimes called an architectural pattern, is a set of principles that provides an abstract framework for a family of systems. An architectural style improves partitioning and promotes design reuse by providing solutions to frequently recurring problems.

2. What are the different types of architectural styles?

a. Data flow styles
   - Batch sequential
   - Pipes and filters

b. Call-and-return styles
   - Main program & subroutines
   - Hierarchical layers
   - OO systems

c. Virtual machines styles
   - Interpreters
   - Rule-based systems
d. Independent components communicating processes styles
   - Event systems

e. Data-centered styles (repositories)
   - Databases
   - Blackboards

3. What are the benefits of using architectural styles?
   - Reuse
   - Understandability of system organization
   - Interoperability
   - Style specificity

4. What is pipe and filter style?
   - Each component has set of inputs and set of outputs
   - A component reads streams of data on its input and produces streams of data on its output.
   - By applying local transformation to the input streams and computing incrementally, so that output begins before input is consumed. Hence, components are termed as filters.
   - Connectors of this style serve as conduits for the streams transmitting outputs of one filter to inputs of another. Hence, connectors are termed pipes.

5. Mention the advantages of pipe and filter style.
   - The pipe-and-filter style simplifies system maintenance and enhances reuse for the same reason—filters stand alone, and we can treat them as black boxes.
   - Both pipes and filters can be hierarchically composed: Any combination of filters, connected by pipes, can be packaged and appear to the external world as a filter.
   - Because a filter can process its input in isolation from the rest of the system, a pipe-and-filter system is easily made parallel or distributed, providing opportunities for enhancing a system's performance without modifying it.

6. Mention the disadvantages of pipe and filter styles.
There is no way for filters to cooperatively interact to solve a problem.

Performance in such a system is frequently poor for several reasons, all of which stem from the isolation of functionality that makes pipes and alters so modifiable; these reasons are listed below:

- Filters typically force the lowest common denominator of data representation (such as an ASCII stream). If the input stream needs to be transformed into tokens, every filter pays this parsing/unparsing overhead.
- If an alter cannot produce its output until it has received all of its input, it will require an input buffer of unlimited size. A sort filter is an example of a filter that suffers from this problem. If bounded buffers are used, the system could deadlock.

7. List out the goal of Call-and-Return Architectures

Call-and-Return architectures have the goal of achieving the qualities of modifiability and solvability.

8. Define Remote procedure call systems. What is its goal?

Remote procedure call systems are main-program-and-subroutine systems that are decomposed into parts that live on computers connected via a network. The goal is to increase performance by distributing the computations and taking advantage of multiple processors.

9. What is layered systems?

Layered systems are ones in which components are assigned to layers to control inter component interaction. In the pure version of this architecture, each level communicates only with its immediate neighbours.

10. Define event based implicit invocation.

Instead of invoking the procedure directly a component can announce one or more events.

- Other components in the system can register an interest in an event by associating a procedure to it.
- When the event is announced, the system itself invokes all of the procedure that have been registered for the event. Thus an event announcement “implicitly” causes the invocation of procedures in other modules.
11. **State the strengths of peer-to-peer style. (MAY/JUNE 2016)** **Advantages**

- Easy and simple to set up only requiring a hub or a switch to connect all computers together.
- You can access any file on the computer as long as it is set to a shared folder.
- If one computer fails to work all the other computers connected to it still continue to work.

12. **Disadvantages of peer-to-peer style. (MAY/JUNE 2016)**

- Security is not good other than setting passwords for files that you don't want people to access.
- If the connections are not connected to the computers properly then there can be problems accessing certain files.
- It does not run efficient if you have many computers, it is best to used two to eight computers.

13. **Differentiate between event based style and publish-subscribe style.**

   In software architecture, publish–subscribe is a messaging pattern where senders of messages, called publishers, do not program the messages to be sent directly to specific receivers, called subscribers, but instead characterize published messages into classes without knowledge of which subscribers, if any, there may be. Similarly, subscribers express interest in one or more classes and only receive messages that are of interest, without knowledge of which publishers, if any, there are.

14. **What is client server style?**

   The client-server architectural style describes the relationship between a client and one or more servers, where the client initiates one or more requests, waits for replies and processes the replies on receipt.

15. **What is pipe and filter style?**

   It provides a single type of component, the filter, and a single type of connector, the pipe. Computational, a filter is a data transformer that reads streams of data through one or more input ports and writes stream of data to one or more output ports.

In call-return styles the components interact by requesting services of other components. Each component in this style provides a set of services through one or more interfaces, and use zero or more services provided by other components in the system.

17. List functions of repository components.

The repository components of a shared-data system carry out a number of functions, including providing shared access to data, supporting data persistence, managing concurrent access to data, providing fault tolerance, supporting access control, and handling the distribution and caching of data values.

18. What is virtual machine?

A virtual machine implements an instruction set for an imaginary machine. Often virtual machines are the under-laying mechanism by which a programming language is executed. It specifies an interface between compiler and a real machine. It is an ordered sequence of layer and each layer services the layer above it.

19. State component of repository style.

a. A central data structure representing the current state of the system.
b. A collection of independent components that operate on the central data structure.

t. List two advantages and disadvantages of layered styles. Advantage:

c. It increases abstraction levels.
d. Changes in a layer affect at most the adjacent two layers.

Disadvantage:

a. Difficult in structuring some system in a layered fashion.
b. Not universally applicable.
UNIT V

DOCUMENTING THE ARCHITECTURE

SYLLABUS: Good practices – Documenting the Views using UML – Merits and Demerits of using visual languages – Need for formal languages - Architectural Description Languages – ACME – Case studies. Special topics: SOA and Web services – Cloud Computing – Adaptive structures

COURSE OBJECTIVE: Be familiar with architectures for emerging technologies.

PART-A

1. What is cloud computing?

Cloud computing, also known as on-demand computing, is a kind of internet-based computing, where shared resources and information are provided to computers and other devices on-demand.

2. What are the characteristics of cloud computing?

- Agile
- Cost
- Device and location independence
- Maintenance
- Performance
- Productivity
- Reliability
- Scalability
- Security

3. What are the different service models in cloud computing?

- Infrastructure as a service
- Platform as a service
- Software as a service
4. What is cloud engineering?

Cloud engineering is the application of engineering disciplines to cloud computing. It brings a systematic approach to the high-level concerns of commercialization, standardization, and governance in conceiving, developing, operating and maintaining cloud computing systems.

5. What is the purpose of Cloud Provider Interface?

A Cloud Provider Interface (CPI) provides an abstraction from an underlying IaaS by defining a set of functions for managing virtual machines life-cycle in which might run an elastic service.

6. What is service oriented architecture?

A service-oriented architecture (SOA) is an architectural pattern in computer software design in which application components provide services to other components via a communications protocol, typically over a network.

7. What are the horizontal layers of service oriented architecture?

- Consumer Interface Layer – These are GUI for end users or apps accessing apps/service interfaces.
- Business Process Layer – These are choreographed services representing business use-cases in terms of applications.
- Services – Services are consolidated together for whole-enterprise in-service inventory.
- Service Components – The components used to build the services, such as functional and technical libraries, technological interfaces etc.
- Operational Systems – This layer contains the data models, enterprise data repository, technological platforms etc.

8. What are the vertical layers of service oriented architecture?

- Integration Layer – starts with platform integration (protocols support), data integration, service integration, application integration, leading to enterprise application integration supporting B2B and B2C.
- Quality of Service – Security, availability, performance etc. constitute the quality of service parameters which are configured based on required SLAs, OLAs.
- Informational – provide business information.
Governance – IT strategy is governed to each horizontal layer to achieve required operating and capability model.

9. What is the good practice for documenting architecture?

Document the relevant views, and then add information that applies to more than one view, thus tying the views together.

10. What is an Architecture Description Language?

Modeling notation to support architecture-based development used to define and model system architecture prior to detailed design and implementation.

11. Define ACME.

Acme is a simple, generic software architecture description language (ADL) that can be used as a common interchange format for architecture design tools and/or as a foundation for developing new architectural design and analysis tools.

12. Give the advantages and disadvantages of ADL. (MAY/JUNE 2016)

ADLs have several advantages as well as disadvantages. One advantage is that they are designed to represent architectures in a formal way. Another advantage is that they often are designed to be readable to both human and machines. A disadvantage is that there is not yet an agreement of what the ADLs shall represent, especially when it comes to the behavior of the system.

13. How do ADL differ from other languages? (MAY/JUNE 2016)

Architecture Description Languages (ADLs) differ from programming languages because programming languages bind all architectural abstractions to a specific point whereas ADL suppress such binding. Architecture Description Languages (ADLs) differ from modeling languages because modeling languages are more concerned with the behavior of the whole system rather than parts whereas ADLs concentrate on the representation of the components.
14. What is ADL?

An architecture description helps manager to form development teams based on work assignments identified, allocate project resources, and to track progress on the project.

15. What is use of object diagram?

Object diagram describe the static structure of a system at a particular time. They can be used to test class diagram for accuracy.

16. What is role of manager in architecture document?

Architecture documentation helps manager to form development teams based on work assignments identified, allocate project resources, and to track progress on the project.

17. Explain concept of view Template.

A view template is the standard organization for a view. The purpose of a view template is that of any standard organization: it helps a reader navigate quickly to a section of interest, and it helps a writer organize the information and establish criteria for knowing how much work is left to do.

18. What are cloud deployment models?

Cloud deployment models are:
- Public cloud
- Private cloud
- Community cloud
- Hybrid cloud

19. List essential characteristic of cloud computing?

On-demand self-service, ubiquitous network access, location-independent resource pooling, rapid elasticity, pay per use.
20. What are disadvantage of UML?

Needs customization through profiles to reduce ambiguity.
Difficult to assess consistency among views.
Difficult to capture foreign concepts or views.
UNIT 1

PART- B

1. List all the common software architecture structures. Explain the component connector Structure.

2. How does the nature of your organization affect the architectures that develop? How do the architectures affect the nature of the organizations?

3. i) Discuss how the software architecture affects the factors of influence.
    ii) Differentiate functional and non-functional requirements.

4. Describe the technical importance of software architectures. Further elaborate on the fact that architecture is a vehicle for stakeholder communication.

5. What allocation structure is as applied to software architectures? Explain three allocation structures in practice.

6. Describe the quality attributes in various categories.

7. Explain the various process recommendations as used by an architect while developing software architectures.

8. Describe the common Software Architecture Structures with neat diagram.

9. Relate the structures to each other and explain kruchten’s four views in detail.

10. Software architecture is often compared to building architecture. What are the strong points of this comparison? What is the correspondence in buildings to software architecture structures and views? What is the weakness of this comparison? When does it breakdown?

11. Define software architecture. Explain the common software architecture structures.

12. Explain how the architecture business cycle works, with a neat diagram
13. Explain how the software architectures affect the factors of influence. Hence or otherwise explain ABC.

14. With the help of a neat block diagram of ABC, explain in detail the different activities which are involved in creating a software architecture.

15. What makes a good architectural process? Give various process recommendations and product recommendations.

16. With an example explain various businesses constrains on software architecture.

17. How to identify constraints for architecture?

18. Define architectural model, reference model, reference architecture and bring out the relationship between them.

UNIT 2

PART- B

1. What is availability? Explain the general scenario for availability.

2. What are the qualities of the system? Explain the modifiability general scenario.

3. What is a quality attribute scenario? List the parts of such a scenario.

4. Distinguish between availability scenarios and modifiability scenarios.

5. Give the scenario portion and possible values for availability, modifiability, performance, security, testability and usability scenarios.

6. With the help of appropriate diagrams, explain the availability scenario and testability scenario in detail.

7. Explain the six part scenarios in detail?

8. a) Explain how faults are detected and prevented.
   b) Write a brief note on design time and run time tactics.

9. Usability is not always given due consideration in architecture design, making usability system goals often difficult to achieve because they are treated as an afterthought. Think of a system where you are familiar with the architecture and try to enumerate the usability tactics, if any, it has employed.

10. What is the goal of modifiability tactics? Discuss the modifiability tactics in detail.

11. Explain the following with respect to tactics: i) Fault recovery ii) Internal monitoring iii) Resource arbitration iv) Resisting attacks.

13. Define Quality Attribute Workshop. Write in detail about the steps involved in QAW.
14. List QAW Results? Mention the benefits of QAW.
15. Explain six part scenarios in detail.
16. Classify the various tactics being used and tabulate how they help to achieve quality attributes in detail.
17. Discuss the concept of documenting quality attributes.

UNIT III

PART-B

1. Explain the architectural view in detail with the available notations.
2. Briefly explain the 4+1 view of RUP.
3. Explain in detail about Siemens 4 views.
4. With an example case study, explain the concept of architectural views.

5. a) What are views? How they serve the architecture with examples?
   b) List the steps in documenting a view for architecture.
6. Explain views with reference to concept, choosing the view and its documentation.
7. Discuss the various notations available to represent various views.

8. Discuss Siemens 4 views in detail.
9. Write in detail about the various architectural perspectives.
10. Compare view and viewpoint with its advantages and disadvantages.
11. Illustrate the types of views with suitable examples.
12. What is View Point? Describe the six core viewpoints in detail.
13. Prepare a case study on your own for choosing the views.

14. Give notations for module view types, component and connector view type, allocation view type.
15. List and explain various types of views. Compare and contrast viewpoint models.

UNIT 4

PART-B

1. With neat diagram, explain data flow architectural style
2. Explain call and return architectural style in detail.

3. Show how different architectural solutions to the same problem provide different benefits with a case study.
4. Identify the problem and find a solution to develop a reusable architecture for instrumentation software.

5. Discuss the strength and weakness for the architecture of mobile robotics case study.
6. Define architecture style. Tabulate the summary of all architectural styles.

7. Consider the case study of building software controlled mobile robot. Describe its challenging problems and design considerations with four requirements. Finally give the solution by layered architecture for all the four requirements.
8. Discuss the importance and advantages of the architectural styles with reference to an appropriate application area.

9. a) Write a note on heterogeneous architectures.

   Discuss the invariants, advantages and disadvantages of pipes and filters architectural style.
10. Explain Shared information styles in detail with neat diagram.

11. Describe all the types of Call-and-return style with neat sketch.

12. Explain implicit invocation style with neat diagram.
13. Describe the various Data flow styles with suitable examples.
14. Illustrate the forces that influence the solutions to the problems based on blackboard pattern.
15. Analyze a case study. Pinpoint its problem and Implement solution for the same using each style.
16. Give various sub style of call and return style.
17. Write short notes on layered style.
18. Explain about interpreter style.

UNIT V

PART- B

1. Explain the documentation of views using UML with neat diagram. Explain the concept of cloud computing in detail.
2. Discuss about the concept of service oriented architecture.
3. Explain the concept of ACME language.

4. What are the suggested standard organization points for view documentation?
5. What are the options for representing connectors and systems in UML?

6. Explain the various types of ADL in detail.
7. Illustrate Architectural Description Languages with suitable example.
8. i) Analyze need for formal languages?
   ii) Point out the Merits and Demerits of Visual languages.
9. Explain the good practices in documenting software architecture.
10. Discuss Adaptive structures in detail
11. i) Define Cloud computing. Explain its types and uses in detail.
    ii) Write a note on Service Oriented Architecture.
12. You are a new hire to a project layout a sequence of documentation you would like to have to acquaint you with your new position.
13. Write short notes on visual languages.
14. Write short notes on informal and formal languages.