BEITC7052  |  Software Architecture
---|---
04  |  02  |  ---  |  04  |  01  |  ---  |  05

### Course Objectives:

- To provide students with a strong foundation in developing large, practical software-intensive applications.
- To train students with sound technical exposure to the concepts, principles, methods and best practices in software architecture.
- To develop the ability among students to learn the details of modeling techniques, design, implementation, deployment, and system adaptation.
- To enable students to choose the right tool for the job at hand and document design rationale.
- To prepare students to gain experiences with examples in design pattern application and case studies in software architecture.

### Course Outcomes:

At the end of the course, students should be able to:

1. Argue the importance and role of software architecture.
2. Recognize major software architectural styles, design patterns, and frameworks.
3. Design software architecture for large scale software systems.
4. Describe various documentation approaches and architectural description languages.
5. Apply architectural patterns to quickly generate architectural alternatives and choose between them.

### Prerequisites:

---

University of Mumbai, Information Technology (semester VII and semester VIII) (Rev-2012) Page 28
This course builds on the study of Object Oriented Software Engineering. We assume fluency with Object Oriented Languages and UML

**DETAILED SYLLABUS:**

<table>
<thead>
<tr>
<th>Sr. No</th>
<th>Module</th>
<th>Detailed Content</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td><strong>Basic Concepts</strong>&lt;br&gt;1.1 Concepts of Software Architecture&lt;br&gt;1.2 Models.&lt;br&gt;1.3 Processes.&lt;br&gt;1.4 Stakeholders.</td>
<td>03</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td><strong>Designing Architectures</strong>&lt;br&gt;2.1 The Design Process.&lt;br&gt;2.2 Architectural Conception.&lt;br&gt;2.3 Refined Experience in Action: Styles and Architectural Patterns.&lt;br&gt;2.4 Architectural Conception in Absence of Experience.&lt;br&gt;2.5 Putting it all Together: Design Processes Revisited</td>
<td>05</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td><strong>Connectors</strong>&lt;br&gt;3.1 Connectors in Action: A Motivating Example.&lt;br&gt;3.2 Connector Foundations.&lt;br&gt;3.3 Connector Roles.&lt;br&gt;3.4 Connector Types and Their Variation Dimensions.&lt;br&gt;3.5 Example Connectors.&lt;br&gt;3.6 Using the connector Framework</td>
<td>06</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td><strong>Modeling</strong>&lt;br&gt;4.1 Modeling Concepts.&lt;br&gt;4.2 Ambiguity, Accuracy, and Precision.&lt;br&gt;4.3 Complex Modeling: Mixed Content and Multiple Views.&lt;br&gt;4.4 Evaluating Modeling Techniques.&lt;br&gt;4.5 Specific Modeling Techniques: Generic Techniques, Domain and Style specific ADLs, Extendable ADLs.</td>
<td>04</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td><strong>Visualization</strong>&lt;br&gt;5.1 Visualization Concepts.&lt;br&gt;5.2 Common issues in Visualization.&lt;br&gt;5.3 Visualization Techniques: Textual Visualization, UML, xADL.</td>
<td>04</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td><strong>Analysis</strong>&lt;br&gt;6.1 Analysis Goals.&lt;br&gt;6.2 Scope of Analysis.</td>
<td>06</td>
</tr>
</tbody>
</table>
| 6.3 Architectural Concern being Analyzed.  
6.4 Level of Formality of Architectural Models.  
6.5 Type of Analysis.  
6.6 Analysis Techniques. |
|---|---|---|
| **7** | **Implementation and Deployment**  
6.1 Concepts.  
6.2 Existing Frameworks.  
6.3 Software Architecture and Deployment.  
6.4 Software Architecture and Mobility. |
| **8** | **Applied Architectures and Styles**  
8.1 Distributed and Networked Architectures.  
8.2 Architectures for Network-Based Applications.  
8.3 Decentralized Architectures.  
8.4 Service-Oriented Architectures and Web Services. |
| **9** | **Designing for Non-Functional Properties**  
9.1 Efficiency.  
9.2 Complexity.  
9.3 Scalability and Heterogeneity.  
9.4 Adaptability.  
9.5 Dependability. |
| **10** | **Documentation**  
10.1 Uses of Architectural Documentation.  
10.2 Views  
10.3 Choosing the Relevant Views  
10.4 Documenting a View  
10.5 Documentation across Views |

**Text Books:**


**References:**


**Term work:** Term work should be based on the Lab experiments and assignments.
Suggested Practical List:
1. Modeling using xADL
2. Analysis – Case study
3. Visualization using xADL
4. Integrate software components using a middleware
5. Use middleware to implement connectors
6. Wrapper to connect two applications with different architectures
7. Creating web service
8. Architecture for any specific domain

Theory Examination:

- Question paper will comprise of 6 questions, each carrying 20 marks.
- Total 4 questions need to be solved.
- Q.1 will be compulsory, based on entire syllabus.
- Remaining question will be randomly selected from all the modules.

Weightage of marks should be proportional to number of hours assigned to each module.