## ANDHRA LOYOLA INSTITUTE OF ENGINEERING AND TECHNOLOGY

**Subject:** **SOFTWARE ARCHITECTURE AND DESIGN PATTERNS**

**Teacher :** **Y.RAJESH** **Academic Year:** **2020-21**
**Year/Semester:** **IV-SEM-I**

**Course Outcomes:**

| | |
|---|---|
| **Factual** | 1) Capable of applying these principles in the design of object oriented systems.<br>2) Demonstrate an understanding of a range of design patterns. Be capable of comprehending a de this vocabulary.<br>3) Be able to select and apply suitable patterns in specific contexts |
| **Conceptual** | Design and implement codes with higher performance and lower complexity |
| **Procedural** | be aware of code qualities needed to keep code flexible |
| **Applied** | Experience core design principles and be able to assess the quality of a design with respect to these |

**Text book References:**

**Text book-1 :** Software Architecture in Practice, second edition, Len Bass, Paul Clements & Rick, Kazman, Pearson Education, 2003.

**Text book-2 :** Design Patterns, Erich Gamma, Pearson Education, 1995.The Unix programming Environment by Brain W. Kernighan &
Rob Pike, Pearson.

**REFERENCE BOOKS:**
1. Beyond Software architecture, Luke Hohmann, Addison wesley, 2003.
2. Software architecture, David M. Dikel, David Kane and James R. Wilson, Prentice Hall ,PTR, 2001
3. Software Design, David Budgen, second edition, Pearson education, 2003
4. Head First Design patterns, Eric Freeman & Elisabeth Freeman, O'REILLY, 2007.

**Contents/Activities:**

| | | |
|---|---|---|
| 1 | **Factual:** Factual knowledge consists of the basic elements students must know to be acquainted with a discipline | • The Architecture Business Cycle.(UNIT-1)<br>• What is Software Architecture(UNIT-1)<br>• Architectural patterns, reference models(UNI<br>• Reference architectures, architectural structur views(UNIT-1) |

| | | |
|---|---|---|
| | | • Software product lines, (UNIT-3)<br>• Building systems from off the shelf compone<br>• Software architecture in future(UNIT-3) |
| 2 | **Conceptual:** Conceptual knowledge consists of the interrelations among the basic elements within a larger structure | • Quality Attributes (UNIT-1)<br>• Achieving qualities(UNIT-1)<br>• Architectural styles and patterns design(UNI<br>• Documenting software architectures(UNIT-1<br>• Reconstructing Software Architectures(UNIT<br>• Pattern Description (UNIT-4)<br>• Organizing catalogs, (UNIT-4)<br>• role in solving design problems(UNIT-4)<br>• Selection and usage. (UNIT-4) |
| 3 | **Procedural:** methods of inquiry, and criteria for using skills, algorithms, techniques, and methods. | • **Analyzing Architectures(UNIT-2)**<br>• Architecture Evaluation(UNIT-2)<br>• Architecture design(UNIT-2)<br>• decision making(UNIT-2)<br>• ATAM(UNIT-2)<br>• CBAM(UNIT-2)<br>• Abstract Factory(UNIT-4)<br>• builder(UNIT-4)<br>• prototype(UNIT-4)<br>• singleton(UNIT-4)<br>• adapter(UNIT-4)<br>• bridge(UNIT-4)<br>• composite(UNIT-4)<br>• facade(UNIT-4)<br>• flyweight(UNIT-4)<br>• Proxy(UNIT-4) |

| | | |
|---|---|---|
| 4 | **Applied:** awareness of one's own learning, control and regulation of cognitive processes, self-knowledge, contextual knowledge, and conditional learning | • A-7E-A case study in utilizing architectural struct<br>• The World Wide Web-a case study in interoperabi<br>• Air Traffic Control-a case study in designing for h availability(UNIT-6)<br>• Celsius Tech- case study in product line developm<br>• Chain of responsibility. (UNIT-5)<br>• command (UNIT-5)<br>• Interpreter(UNIT-5)<br>• iterator(UNIT-5)<br>• mediator(UNIT-5)<br>• memento(UNIT-5)<br>• observer(UNIT-5)<br>• state(UNIT-5)<br>• Strategy(UNIT-5)<br>• template method(UNIT-5)<br>• visitor(UNIT-5) |

## Schedule and Sequence:

| Day | Topic<br><br>Unit-1<br>Chapter -1 | Objectives | Before Class-Videos, e-Books, Case Studies, Text book | In-Class-Activities, Quiz |
|---|---|---|---|---|
| 1 | What Is Software Architecture, What Software Architecture Is and What It Isn't | To understand interrelationships, principles and guidelines governing architecture and evolution over time. | T1 | Case Study 1: Key Word in Context app on Shared Memory, Events, ADT, and Dataflow styles |
| 2 | Architectural Structures and Views | | T1 | Case 2: A Fresh View of Compilers app on heterogeneous architectures |
| 3 | What Makes a Good Architecture, Why IsSA Important, Where Do Architectures Come From | | T1 | Case 3: A Layered Design with Differen Styles for the Layers |
| 4 | Software Processes and the ABC | | T1 | |
| 5 | Architectural Patterns, Reference Model, and | | T1 | |

| | | | | |
|---|---|---|---|---|
| | Reference Architecture. Pattern System – What is Pattern System, Pattern Classification, Pattern Selection | | | |
| **6** | Introduction to Architectural Patterns, Pipes & Filter | | T1 | Case 4: An Interpreter Using Different Idioms for the Components |
| **7** | Model-View-Controller | | T1 | Case Study 1: Key Word in Context app on Shared Memory, Events, ADT, and Dataflow styles. |
| | **Chapter-II** | | | |
| **1** | Understanding the Requirements – Functionality & Architecture, Architecture & Quality Attributes (QA), System QAs | | T1 | J2EE/EJB framework: A Case Study of an Industry- Standards Computing Infrastructure. |
| **2** | QA Scenarios in Practice | | T1 | |
| **3** | Business & Architecture Qualities | | T1 | |
| **4** | Achieving Qualities | | T1 | |
| **5** | Designing the Architecture – Architecture in the Life Cycle | | T1 | ASP.NET MVC 3 /4 frameworks [**Reference:** ASP.NET MVC - http://www.asp.net/mvc http://pluralsight.com/training /Player?author=scott-allen&name=mvc4-building- m6-ajax&mode=live&clip=0&course=mvc4-building] |
| **6** | Attribute Driven Design | | | |

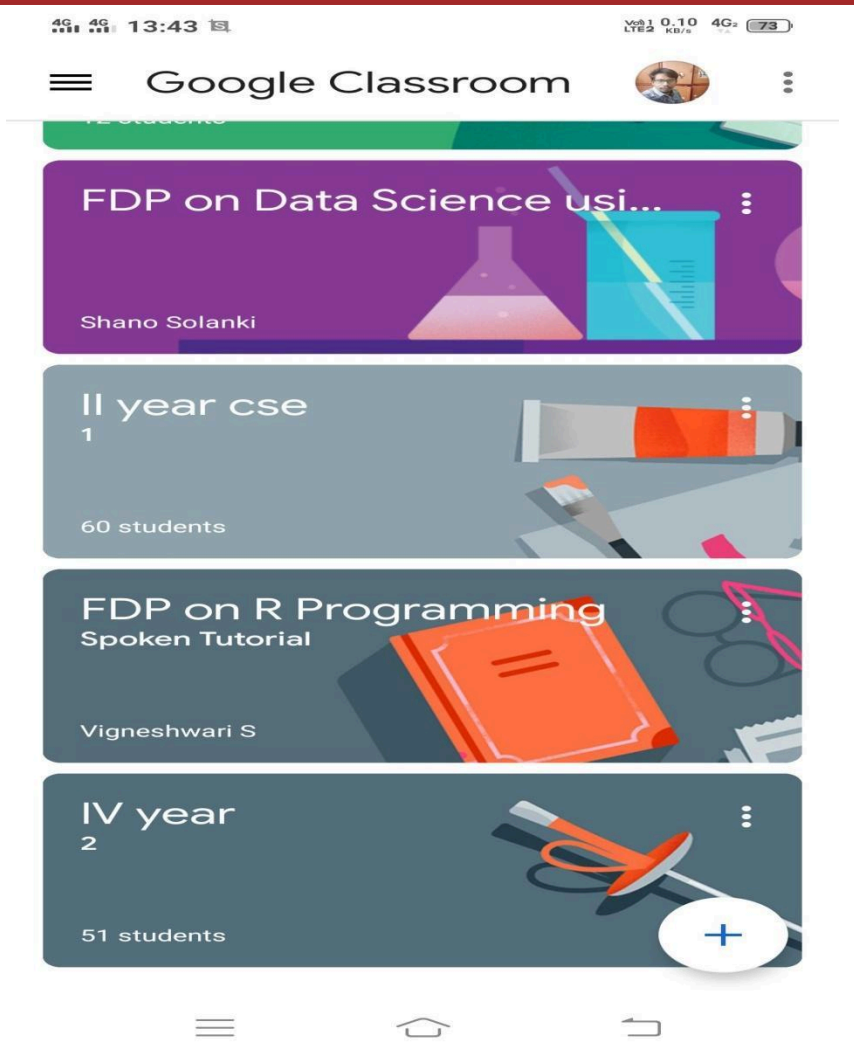
| | | | | |
|---|---|---|---|---|
| | (ADD) | | T1 | |
| 7 | Documenting Software Architectures – Uses of Architectural Documentation, Views | | T1 | |
| 8 | Reconstructing Software Architectures – Information Extraction, Database Construction, View Fusion, and Reconstruction. | | T1 | |
| | **UNIT-2 Evaluating & Reconstructing the Architecture and Moving from one system to many** | To understand various architectural styles of software systems. | T1 | |
| 1 | **Evaluating the Architecture – The ATAM** | | T1 | Comparison of ATAM,CBAM,SAAM |
| 2 | CBAM | | T1 | |
| 3 | Moving from one system to many – Software Product Lines | | T1 | |
| 4 | Building Systems from off the shelf Components | | T1 | |
| 5 | Software Architecture in the future | | T1 | |
| | **UNIT-3 Introduction to Design Pattern and Creational Patterns** | | | |

| | | | | |
|---|---|---|---|---|
| 1 | **Introduction to Design Patterns**<br>What is Design Pattern(DP) | To understand design patterns and their underlying object oriented concepts. | T2 | Write a Java Program on Singleton<br><br>A Case Study: Design a Document Editor using Lexi **[Text Book 2]]** |
| 2 | Design Patterns in Smalltalk MVC | | T2 | |
| 3 | Describing DPs | | T2 | |
| 4 | The Catalog of DPs & Organizing the Catalog | | T2 | |
| 5 | How DP Solve Design Problem & How to Select & Use of a DP | | T2 | |
| 6 | **Creational Patterns:**<br>Abstract Factory &Builder | | T2 | |
| 7 | Factory Method & Prototype | | T2 | |
| | **UNIT – IV AND V : Structural and Behavioral Patterns** | To understand implementation of design patterns and providing solutions to real world software design problems. | | |
| | **Structural Patterns:** Adapter | | T2 | Write the 14 elements for the following Patterns: Bridge, Façade, Proxy, Chain of Responsibility, Interpreter, Memento, Strategy, Visitor |
| | Composite & Decorator | | T2 | |

| | | | | |
|---|---|---|---|---|
| | Flyweight | | T2 | |
| | **Behavioral Patterns:** Command | | T2 | |
| | Iterator & Mediator | | T2 | |
| | Observer & State | | T2 | |
| | | | | |
| | | | | |
| | **UNIT-6 Case Studies** | To understand patterns with each other and understanding the consequences of combining patterns on the overall quality of a system. | | |
| | The World Wide Web - a case study in Interoperability, | | T1,T2 | |
| | Air Traffic Control – a case study in designing for high availability, Celsius | | T1,T2 | |
| | A-7E – A case study in utilizing architectural structures, | | T1,T2 | |
| | **A Case Study (Designing a Document Editor):** Design Problems, Document Structure, Formatting, Embellishing the User Interface, Supporting Multiple Look-and-Feel Standards, Supporting Multiple Window Systems, User Operations, Spelling Checking and Hyphenation. | | T1,T2 | |
| | | | | |

## Evaluation:

- Formative Assessment – 50%
- Summative Assessment – 50%




Prepared: Faculty

Verified: HOD

# ANDHRA LOYOLA INSTITUTE OF ENGINEERING AND TECHNOLOGY

Teacher/Instructor:Mr.Y.RAJESH

Department of COMPUTER SCIENCE AND ENGINEERING

Assistant Professor of CSE

**MICROLESSONPLAN(ACCORDINGTOBLOOMSDIGITALTAXONOMY)**

| | |
|---|---|
| **Programme** | B.Tech,Computer Science and Engineering |
| **Semester** | IVYear-I Semester |
| **SubjectTitle** | Software architecture and design pattrens |
| **SubjectCode** | R1641052 |
| **ClassHours** | 5-Hoursperweek |
| **TotalHours** | 70 |
| **Credits** | 3 |
| **MaxMarks** | 100 |
| **Unit&Title** | Un |
| **TeachingandLearning** | BlackBoard/PowerPointPresentation/Videos,E-material. |

| **Detailed – Lesson**<br><br>**LessonObjectives:** | |
|---|---|
| Factual | 4) Capable of applying these principles in the design of object oriented systems.<br>5) Demonstrate an understanding of a range of design patterns. Be capable of comprehending a design presented using this vocabulary.<br>6) Be able to select and apply suitable patterns in specific contexts |
| Conceptual | Design and implement codes with higher performance and lower complexity |
| Procedural | be aware of code qualities needed to keep code flexible |
| Applied | Experience core design principles and be able to assess the quality of a design with respect to these principles |

**Prerequisite Knowledge:**

1.know about the uml diagrams

2. students are know to construct a uml models

3.how to construct the all the uml models.

**MicroLessonPlan:**

1. **Pre-taskActivity-Introducing the UML model**

   Architectural Patterns, Reference Model, and Reference Architecture.

   Pattern System – What is Pattern System, Pattern Classification, Pattern Selection

2. **In-classActivity**:
   **Component-and-connector structures**
   **Elements are runtime components (units of computation) and connectors (communication vehicles among components)**
   **The relation is attachment, showing how the components and connectors are hooked**

3. **together**
   - **What are the major executing components and how do they interact?**
   - **What are the major shared data stores?**
   - **Which parts of the system are replicated?**
   - **How does data progress through the system?**
   - **What parts of the system can run in parallel?**
   - **How can the system's structure change as it executes?**
   - **Process, or communicating processes**

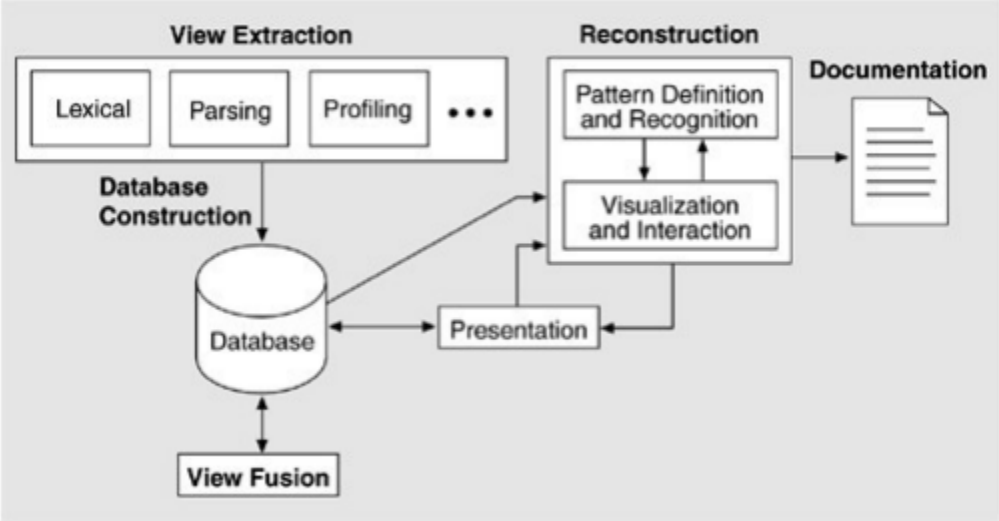
- units are processes or threads that are connected with each other by communication, synchronization, and/or exclusion operations
- Concurrency
- The units are components and the connectors are “logical threads”
- A logical thread is a sequence of computation that can be allocated to a separate physical thread
- Shared data, or repository
- This structure comprises components and connectors that create, store, and access persistent data
- Client-server
- The components are the clients and servers, and the connectors are protocols and messages
- Allocation structures

The relationship between the software elements and the elements in one or more external environments

- What processor does each software element execute on?
- In what files is each element stored during development, testing, and system building?
- What is the assignment of software elements to development teams?
- Deployment
- Shows how software (usually a process from a component-andconnector view) is assigned to hardware-processing and communication elements
- Relations are “allocated-to” and “migrates-to” if the allocation is dynamic
- Implementation
- how software elements (usually modules) are mapped to the file structure(s)

- **Work assignment**
- **assigns responsibility for implementing and integrating the modules to development teams.**

## 4. simple problem:




## 5. Post–taskActivity:

InPosttaskactivityrevisingtheclass,clarifyingthedoubtsandaskingquestionstoknowtheresponse.

Question1:

1. What is architecture?

Question2:

2. Types of architecture?

Question3:

3. Explain about reconstruction technique.

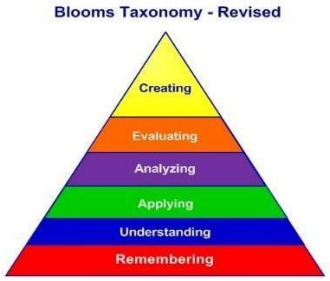
## 6.Discussion

- Students will be able to remember the concept of architecture

## 7. references:

**Text book-1 :** Software Architecture in Practice, second edition, Len Bass, Paul Clements & Rick, Kazman, Pearson Education, 2003.

**Text book-2 :** Design Patterns, Erich Gamma, Pearson Education, 1995.The Unix programming Environment by Brain W. Kernighan & Rob Pike, Pearson.




## TaxonomyofObjectives –SpecificObjectives

| Knowledge Dimension | TheCognitiveProcessDimension | | | | | |
|---|---|---|---|---|---|---|
| | **Remember** | **Understand** | **Apply** | **Analyze** | **Evaluate** | **Create** |
| A.Factual Knowledge | **SO-1** | | | | | |
| B. Conceptual Knowledge | | S0-2 | | | | |
| C. Procedural Knowledge | | | SO-3 | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| D.Meta Cognitive Knowledge | | | | SO-4 | | |