

Design Patterns – [7076]

Course Introduction:

A design pattern provides a general reusable solution for the common problems that occur in software design. The patterns typically show relationships and interactions between classes or objects. The idea is to speed up the development process by providing well tested, proven development/design paradigm.

Course Prerequisite: OOP

Course Learning Outcomes:

- To this end, students will learn and gain hands-on experience in designing software systems by reusing/applying design patterns.
- Design patterns are successful solutions to recurring problems that arise when building software systems.
- Reusing design patterns helps to prevent subtle issues that can cause major problems and improves code readability for the developers familiar with the patterns.
- In addition to mastering these good design abstractions, the students will also learn how to evaluate a design, identify common problems, and how to fix these problems through refactoring.

Course Plan

Week	Topic
01	Introduction to OOD & UML
02	Introduction to Design Patterns, Types of Patterns Creational Pattern – Singleton Implementation & Use cases
03	Creational Pattern – Factory Implementation & Use cases
04	Creational Pattern – Builder Implementation & Use cases
05	Structural Pattern – Adapter Implementation & Use cases

06	Structural Pattern – Decorator Implementation & Use cases
07	Behavioral Pattern – Observer Implementation & Use cases
08	Behavioral Pattern – Mediator Implementation & Use cases
09	Behavioral Pattern – State Implementation & Use cases
10	Design Methodologies & Architecture
11	Design Methodologies & Architecture
12	Project Discussion
13	Project Viva
14	Project Viva

Grading:

- Class Participation: 30%
- Assignments: 20%
- Mid: 20%
- Final: 30%

Instructor: Aneeq Hashmi